

1JL2736

MULTIRESOLUTION SEGMENTATION METHODOLOGY
FOR RESPIRATORY ELECTROMYOGRAPHIC SIGNALS

By

HAAN-GO CHOI

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1992

*"The fear of the LORD is
the beginning of knowledge"*
(Proverbs 1:7)

ACKNOWLEDGEMENTS

The author would like to thank the members of his supervisory committee for their guidance and assistance in doing this work. In particular, the author wishes to express deep gratitude to his advisor, Dr. Jose C. Principe, for his continual guidance, encouragement, and support through the course of this research.

The author would like to gratefully acknowledge the collaboration of Dr. Alastair A. Hutchison and Dr. John A. Wozniak of the Department of Pediatrics, College of Medicine, for spending their precious time scoring the EMG signals and for their helpful discussions. He is also grateful to Dr. Jack R. Smith, Dr. John Staudhammer, and Dr. Paul W. Davenport for their interest and helpful comments.

In addition, the author would like to thank his colleagues of the Computational Neuroengineering Laboratory for their moral support, encouragement, and advice: Taehwan Yoon, Chongtai Kim, Seunghun Park, Jingeol Lee, Russell, Bert, Jeff, Armando, Kuo, Karl, and Samel. He also thanks Laura Higgs for editorial correction.

Finally, special appreciation is expressed to his wife, Kyung-Ae, and children, Da-Jung and Jun-Young (Edwin), for their prayers, patience, and love; without them he would never have had the strength or courage to complete the program.

This work was partially supported by the grant from NIH HL39858.

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGEMENTS	iii
ABSTRACT	vii
CHAPTERS	
1 INTRODUCTION	1
1.1 Electrophysiological Basis of Electromyogram.....	1
1.2 Respiratory EMG.....	3
1.3 Statement of the Problem.....	7
1.4 Related Research.....	10
1.5 Objectives.....	12
1.6 Overview	14
2 SIGNAL SEGMENTATION METHODOLOGY	16
2.1 Introduction.....	16
2.2 Overview of Segmentation Model.....	19
2.3 Preprocessing	23
2.3.1 Noise Sources of Respiratory EMG Signals.....	23
2.3.2 Noise Rejection.....	25
2.3.3 Design of Bandpass Filter.....	27
2.3.4 Design of an Adaptive Notch Filter	32
2.4 Burst Activity Identification and Signal Segmentation	46
2.4.1 Burst Activity Identification	46
2.4.1.1 Signal averager - Moving Time Average (MTA).....	46
2.4.1.2 Relationship between flow and EMG signals.....	51
2.4.2 Signal Segmentation	54
2.5 Detection of Boundary Candidates.....	58
2.5.1 Signal Enhancement - Short Time Energy (STE).....	58
2.5.2 Combination of MTA and STE	60
3 BOUNDARY ESTIMATION USING ARTIFICIAL NEURAL NETWORK.....	63
3.1 Introduction.....	63
3.2 Neural Network as Pattern Classifiers	66
3.2.1 Posterior Probability Estimation	67

3.2.1.1 Classification	67
3.2.1.2 Least squares error measure for posterior probability estimation.....	67
3.2.1.3 Conditions as probability estimation	70
3.2.1.4 Neural net as a discriminant.....	71
3.2.2 Network Learning Algorithm	74
3.3 Preprocessing for the Network Input	80
3.3.1 Peak-to-Peak Detection.....	81
3.3.2 Line-length Computation (Duration-Slope).....	83
3.3.3 Amplitude Normalization	85
3.3.4 Quantization of Duration-Slope.....	89
3.4 Factors in Implementing the Neural Network	90
3.4.1 Number of Input Pattern	91
3.4.2 Input Quantization	91
3.4.3 Training Sets	92
3.4.4 Network Architecture	95
3.5 Determination of the Final Boundary.....	97
 4 SIGNAL PROCESSING AND VISUALIZATION SYSTEM.....	 100
4.1 Introduction.....	100
4.2 Environment of System Development	102
4.3 General Signal Algorithm Processing and Visualization Tool - " <i>Signal Editor</i> "	108
4.3.1 Overall System Architecture.....	109
4.3.2 Hierarchical Structure of Windows	114
4.3.3 Functional Description and Visualization.....	116
4.4 Implementation of the Segmentation System	126
4.4.1 Overall System Architecture	127
4.4.2 Functional Description.....	130
4.4.3 Signal Visualization	139
4.5 Discussions	146
 5 SYSTEM EVALUATION AND RESULT	 148
5.1 Data Acquisition and Transfer	148
5.2 Evaluation Procedures	150
5.3 Test Results	153
5.3.1 Graphical Presentation	153
5.3.2 Numerical Presentation	160
5.3.2.1 Evaluation of the transitional segments	161

5.3.2.2 Boundary comparison between humans and the system	167
6 CONCLUSION.....	174
6.1 Summary of Main Ideas.....	174
6.2 Future Work	178
REFERENCES.....	180
BIOGRAPHICAL SKETCH.....	185

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

MULTIRESOLUTION SEGMENTATION METHODOLOGY
FOR RESPIRATORY ELECTROMYOGRAPHIC SIGNALS

By

HAAN-GO CHOI

DECEMBER 1992

Chairman: Dr. Jose C. Principe
Major Department: Electrical Engineering

The goal of this dissertation is to develop an objective segmentation methodology for the respiratory EMG signals, implement the method in a computer, and evaluate the system's performance.

The segmentation model concentrates on estimating the boundaries such as onset and cessation of a burst activity. Segments of the clear burst activity are discriminated from background segments by relying on multi-channel information for robust detection. A new segment, the transitional segment of variable length in which the boundaries are located, is proposed. Within the transitional segment, half-waves that may contain the final boundary point are called boundary candidates. For final boundary estimation, an artificial neural network ranks the half-waves by their posterior probabilities. The final boundary point for the given transitional segment is detected within the half-wave of the highest probability. The functional flow of the segmentation model is multi-resolution analysis because the scope of signal analysis is narrowed from the signal segment to the sample.

In this dissertation, this signal processing methodology was implemented in a computer. One part is a general purpose signal analysis environment to help the

development of digital signal processing algorithms. The user-friendly, window-based tool enables the user to simultaneously visualize and measure parameters for any kind of multichannel data. The other is an automated segmentation system for the boundary detection of respiratory EMG signal, in which the segmentation model is implemented and evaluated. Data from five animals (lambs) are used for the evaluation.

Results are presented graphically and numerically. Graphical results show the transitional segments, boundary estimates with probabilities, and final boundaries. The system-detected boundaries are compared with two humans to assess the system performance. The humans-detected boundaries are found in the computer defined coarse transitional segments (47.1 ± 1.2 [msec]) and most of them (545 out of 552 boundaries, i.e., 98.73%) are found in the fine transitional segments (22.7 ± 0.8 [msec]).

Overall, the variability between the system- and either human-detected boundaries is less than the variability between the humans.

CHAPTER 1

INTRODUCTION

Electromyography (EMG) is the study of muscle function through the examination of the electrical signals the muscles produce [Ba85]. The measurement of muscle-generated electricity and the relationship between electricity and muscle contraction were observed as early as the 17th century. During the past two centuries, a wide variety of research has contributed to the understanding of the electrical phenomenon which is associated with muscle contraction.

With the introduction of high technological electronic devices, the task of collecting and analyzing the EMG signal has been greatly simplified. Since electromyography has been put to use in the clinical environment, sophisticated signal analysis is required. Thus, it is quite natural that applying electric currents to muscles, detecting electric signal from muscles, and analyzing the signals have attracted the attention of the biomedical engineers.

Myoelectric signals detected from the muscles have been used in various applications, such as kinesiology, myoelectric control of powered prostheses [Ja82], control of functional electrical stimulation for the paralyzed muscles [Vo81], biofeedback training [Bas75], the study of neurological diseases [Ki83], and analyses for diagnostic aids [Ge84].

1.1 Electrophysiological Basis of the Electromyogram

Each nerve fiber in a motor nerve makes electro-chemical connections with several muscle fibers in its target muscle. Normally one nerve action potential arriving at a neuromuscular junction gives rise to one post-junction muscle fiber action potential. Depolarization of the muscle fiber by the neurotransmitter released from the nerve terminal

causes local ionic currents through the membrane and in the interstitial space. These currents lead to a large local change in muscle fiber transmembrane potential and also initiate depolarization at neighboring sites on the membrane. Electrodes placed in the extracellular space near the muscle fiber record a potential difference that arises from the flow of these ionic currents.

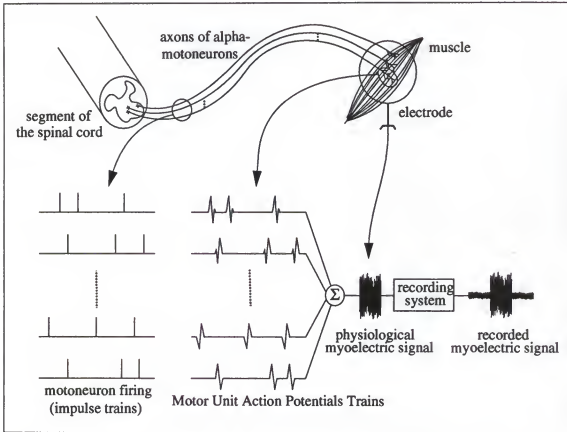


Figure 1.1. Schematic representation of the generation of the myoelectric signal

The electromyogram (EMG) is the signal obtained by the electrical activity of a muscle. The upper part of Figure 1.1 depicts the essential physiological process of EMG measurement: the muscle fibers are activated by alpha-motoneurons located in the spinal cord via their axons which form the nerve bundle. One single neuron stimulates a muscle-fiber group (motor unit, MU) that gives rise to a specific waveform, called a motor unit action potential (MUAP). The shape of the measured MUAP depends on the geometry of

the needle electrode used and its position relative to the muscle fibers. The repetitive innervation of several individual motor units results in a superposition of pulse trains (MUAPTs = Motor Unit Action Potential Trains), which then constitute the EMG. Distinct MUAPs can only be analyzed during weak contractions when few motor units are active [Le82 and Gu83]. During strong contractions, numerous MUAPs are randomly generated and superimposed, and the EMG becomes a noise-like “interference” signal.

The electrical activation of a muscle can be assessed by recording the electrical activity in the muscle itself or in the motor nerve to the muscle. EMGs are usually recorded because electroneurograms (ENGs) are difficult to obtain. One might assume that the ENG from a motor nerve and the EMG from its associated muscle should be “equivalent.” However, because of the rapid attenuation of the electrical activity with increasing distance of the electrical source from the electrodes, EMGs sample only a part of the population of contracting muscle fibers. On the other hand, the ENG samples essentially all active fibers in a nerve. In ENG recordings, then, the recorded activity is weighted primarily by the contribution of each action potential voltage amplitude, but in the EMG the distance of the fiber from the electrodes is also an important factor. In addition, one nerve fiber action potential gives rise to several muscle fiber action potentials. Nevertheless, mass discharge activities of a motor nerve and its muscle show their correlation [Bru84, Lo66, and De79].

1.2 Respiratory EMG

Figure 1.2 shows the major respiratory organs and their signals. Respiratory muscle activities determine the opening and closing of the laryngeal valve. The driving pressure generated by the pump muscle is related in time to the degree of mechanical opening and closing of the valve, and thus to the airflow in and out of the lungs. In the presence of pump muscle activity, the more intense the activity of the abductor laryngeal muscle, the greater the inspired airflow, whereas, in the absence of pump muscle activity, the greater the expiratory airflow. The more intense the activity of the adductor laryngeal muscle, the

greater the closure of the valve and the greater the decrease in airflow. Therefore, the respiratory airflow pattern and lung volume are determined by respiratory muscle activities (specifically, the pump muscles and those of the laryngeal valve). In addition, the timing of the muscle activities is a key determinant of the airflow pattern. Indeed, this aspect is critical in the control of lung volume. The main pump muscle is the diaphragm (DEMG). The only laryngeal or upper airway muscle which opens the vocal cords is the PosteriorcricoArytenoid (PCA) muscle. Inspiratory airflow is strongly related to these two muscle activities.

Respiratory muscle activities are generated by the integrated neural motor output of the central nervous system. This neural output is determined centrally by intrinsic processes and modified by afferent information. Currently the measurement of the respiratory neural output is not easy and the quality of recording is poor. However, the respiratory muscle EMG has been shown to reflect this neural output. Thus, information derived from the respiratory muscle EMG can provide insight into the control of respiratory patterns and absolute lung volume. Such knowledge is important to the clinician's understanding of respiratory problems as shown by the presence or absence of an abnormal respiratory pattern and/or abnormal activities of the respiratory muscles. Furthermore, it has been shown that, in the newborn, information derived from the study of the respiratory muscle EMGs gives insight into both factors which determine the establishment of lung volume after birth and the pathophysiology of asphyxia and its treatment [Ko88].

Figure 1.2(b) shows respiratory signals. The first signal is volume, the integration of flow shown in the second signal. The other signals are PCA and DEMG. The interval, $T1$, indicates the period of a breath, and intervals, $T2$ and $T3$, indicate durations of inspiration and expiration. The interval, A , indicates the duration of a burst activity of the muscle and B indicates the noise contaminated background (or called the baseline). Frequently the background contains a tonic activity, continuous low-level muscular activity which is just above pure noise. It comes from an incompletely relaxed contraction of the

muscle. Points, O and C , in the figure indicate points at which the muscle starts to contract or release. The point, O , is called an onset point (onset boundary or simply onset) of the burst activity while the point, C , is called a cessation (or return-to-baseline) point. Intervals, T_o and T_c , onset and cessation transitional segments, respectively, indicate intervals in which the onset and cessation boundaries are located. The relationship between the four signals shows consistent patterns.

If the clinician is going to understand the basis of a pathological breathing pattern and its changes with breathing disorders, he must be able to analyze the activities of respiratory muscle EMGs as well as their relationships to one another and to spirometric measurements such as tidal flow, volume, and changes in airway pressure. Thus, accurate and objective methods of EMG signal analysis are required to characterize respiratory muscle activities in order to comprehend the airflow pattern and its brain control, especially in terms of precise timing measurements. Examples of such measurements include timing differences between PCA onset and DEMG onset, between PCA cessation and DEMG onset, between inspiration of flow and PCA onset, etc.

Thus, the problem of detecting the boundaries (sometimes called endpoints), i.e., onset and cessation points of muscle burst activity from the nonactivating background or noise, is important in respiratory EMG signal analysis. In particular, the problem of locating other signal landmarks (e.g. peak and offset) during the period of the burst activity is inherently based on the assumption that one can locate the region of the muscle burst activity to be detected. To accomplish the timing measurements, at least the following timing points should be detected: peaks and zero-crossings of volume and flow signals, and onset, cessation, and maximum energy points of the burst activity of PCA and DEMG signals. The signal analysis of volume and flow is simple. However, the difficulty is in the detection of the onset and cessation points of muscle signals due to the fuzziness of signal changes. The major task of this research is to develop a robust and automatic method to detect the boundary points of the burst activity.

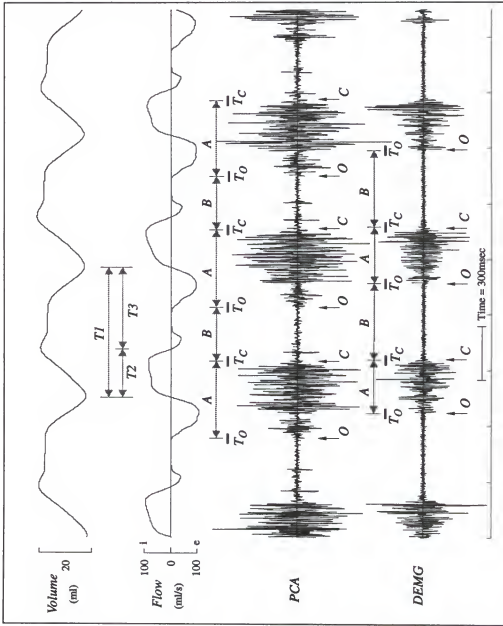


Figure 1.2. Description of signals for respiratory pattern study

1.3 Statement of the Problem

Except in the case of physiological environments with an extremely high signal-to-noise ratio, boundary detection is not a trivial problem. For such cases, the energy of the lowest-level muscle activity (e.g., slow and weak neuromotor response) exceeds the background energy, and a simple amplitude or energy threshold suffices to discriminate the onset and cessation points from the burst activity. However, such ideal recording conditions are not common for real-world applications. Thus, simple energy measures are not sufficient for separating the weak muscle activity from the background. Nevertheless, analog integration or digital averaging methods have been widely utilized for the analysis of interference respiratory muscle signals [Bru84, Ko88, Ne90, Sm89, St86, and Clb90].

Since the average signal smooths out the EMG signal, the accurate setting of the threshold to discriminate the burst activity from the background segment is crucial. Based on the currently available respiratory EMG data, it is known that a mixture of sparse MUAPs and/or tonic activity may occur in the background segment. The rejection of these sparse activation potentials and the reduction of the tonic activity effect would make the recognition of the burst activity and background segments much easier. Actually the suppression of these activities is difficult because they share same characteristics with the burst activity. The use of an averaged signal, such as moving time average or Reimann sums [Ne90] derived from the raw EMG signal, provides a clear indication of the changes in slope between the background and the burst activity.

However, the value of the threshold is a painstaking task because the threshold is calculated relative to the background segment (thresholding segment) which is inherently quite variable in the interference EMG signal. Such a test often requires that the threshold itself be normalized to a general intensity level; if the burst activity is weak, a lower threshold will have to be used. Alternatively, the intensity itself may be normalized. Thus, a simple amplitude comparison technique cannot segment precisely all kinds of respiratory muscle signals in a consistent manner. Moreover the boundary detection using the averaged

signals does not provide the necessary accuracy.

The averaging technique is useful for the characterization of slowly changing muscle signals with a stable background because its output represents the muscle activity envelope. For a rapidly changing muscle signal, the output of the averaged signal does not follow the muscle activity envelope. Instead, it smooths the muscle signal. Therefore, there are errors in the detected boundaries of rapidly changing muscle signals when processed within parameters that work well for slowly changing muscle signals.

The test signals which are used in the research show these phenomena; usually the slow signal changes in onset transitional segments and the rapid signal changes in cessation segments. Consequently, when a constant thresholding method is used, the best value of the threshold for the boundary detection for one cycle of breath is not necessarily the best for another cycle of breath. The same problem arises when the same thresholding method is applied to other muscle signals of the same subject or those collected from other subjects because the signal shapes of the respiratory EMG are highly variable. Therefore, one of the most important factors affecting the performance of the boundary detection is the adaptive formulation of the threshold when the averaging technique is used.

Major difficulties in analyzing the EMG signals, especially locating boundary points, are explained by pointing out three problems. First, the EMG signal generally can be referred to as nonstationary. The statistical characteristics of the signal are not always the same when the EMG signal is measured at different times. Second, the interference EMG signals have a highly variable signal shape in the transitional segment near the onset and cessation of the burst activity when the associated muscle contracts and relaxes. Third, there is no precise definition or benchmark to decide which point really represents a boundary. Conceptually, the boundary can be defined as the point crossing the background level. However, it is hard to establish a universal signal analysis methodology to detect the boundary.

Depending upon the distance from the electrode, strong and weak signals from

close and distant single muscle fiber action potentials are combined irregularly so that the pattern of this interference muscle signal is quite different from breath to breath. Moreover, each muscle burst activity usually starts and ends gradually, although it can occasionally be abrupt. In other words, the boundaries between burst activation and nonactivation of the muscle signals do not always show a sharpness in amplitude, resulting in boundary decision ambiguity. These facts mean that the boundary definition may not always be clear from the physiological point of view. The precise definition of the boundary may be also a controversial subject among EMG experts.

Thus, the crucial problems in boundary location are the vagueness of the boundary definition and the signal fuzziness around the boundary points. The research issue is how to tackle the boundary detection problem for highly variable signal patterns in order to enhance detection accuracy and reduce variance.

Spectrum (or frequency) analysis, which has been the most frequently used method for describing biological signals, transforms the time domain data into the frequency domain, with the purpose of enhancing the characteristics of the signal. There are three main considerations for the boundary detection with the frequency domain approach. (1) The spectral analysis description cannot be used directly in the subsequent interpretation stage since the medical and clinical knowledge which has been accumulated so far is mainly based on the human expert's visual discrimination. Moreover, the spectral analysis is known to be inappropriate for describing aperiodic or irregular activities, which commonly occur in the interference respiratory EMG signal. (2) Frequency domain analysis is useful when the spectrum of the desired signal does not overlap that of the background noise spectrum. In the case of the respiratory muscle signal, the frequency distribution of the burst activity is quite broad, overlapping that of the background segment. That is, it does not have dominant frequency components. (3) Spectrum analysis techniques may detect a change in spectral content between two observed windows, but the resolution in time is bounded by the window length. This comparison of window spectrum cannot

provide the necessary temporal accuracy.

Because the objective of this research requires both the recognition of the burst activity and the background as well as precise boundary detection, the time domain approach is selected based on the fact that rough discrimination between the burst activity and the background segment can be done directly by amplitude and/or energy comparison. However, frequency analysis will be used for signal preconditioning. The EMG signal will be preprocessed to eliminate noise contamination generated by electrical interferences and low frequency components due to the motion movement.

1.4 Related Research

Several attempts have been made to automate the detection of the EMG signal from the background noise. The problems of segmenting a signal in stationary intervals or detecting abrupt changes in a signal frequently arise in various modeling and processing methods for nonstationary digital signals. Such methods are required in speech processing [Ra78, La81, Gr80, and Ha77], image processing [Bas81], acoustics [Eg89, Wo38, and Bas83], and in the automatic analysis of biomedical signals, such as electrocardiograms (ECG) [Gus78], electroencephalograms (EEG) [Bo77, Is81, and Ish80], and EMG signals [Ne90, Ya80, Ya82, Sm89, and Clb90]. An analysis of the behavior of these signals reveals that most of the abrupt changes that occur are either changes in the mean signal level, as in the case of edge detection in digital pictures and for some geophysical signals and the EMG, or changes in the spectral characteristics of the signal, as in speech and EEG signals.

There have been many studies on acoustic segmentation of speech with results ranging from simple speech detection algorithms to classification algorithms. The major features adopted in the speech research [Si82, Sa88, Wil84] are the zero-crossing rate, signal energy, the correlation between adjacent speech samples, linear predictor coefficients or error, and autocorrelation coefficients. The speech signal is classified into voiced and unvoiced segments. In other applications such as the classification of radar

signals, spectral transition and adaptive autoregressive filtering (feature vector extraction) have been utilized. The basic assumption is that many of the nonstationary signals representing physical phenomena are piecewise stationary. That is, although the signal statistics may vary significantly over the complete data record, localized regions can be identified where the statistics remain approximately constant. As an example, the pitch, roll, and yaw segments are classified from the radar signal.

In EEG signal analysis [Got87, Ca72, Pri85, and Ko85], bursts of spike-and-wave activity and epileptic seizures have been detected. The researchers used derivatives and a slow wave detection method based on time-domain characteristics such as slope, the duration between zero-crossings, and slow wave amplitude in one or several signal channels. In addition, they have used various criteria of spatial spread or temporal succession to perform the detection.

In respiratory EMG signals, an objective analysis of the interference EMG signals is a complex task. Unfortunately, little research has dealt with the method of boundary detection. Most published papers describe conceptual approaches. The principal feature of identifying the burst activity from the background is amplitude or energy. The most commonly used method relies exclusively upon detecting points of signal changes or changes in signal slopes on the averaged EMG signals. The slope measurement used to detect the boundaries is based upon the degree of change in the average signal relative to the level of the background segment. That is, it compares the energy with some threshold value and identifies the onset of the burst activity at which the energy exceeds the threshold and the cessation point at which energy drops below the threshold. This technique has proven useful because the average signal correlates with other physiological measurements [Bru84, Gr80, and Ha77].

This averaging algorithm, associated with an analog integrator, has an inherent advantage: it provides an accurate averaged value which is proportional to the muscle force while reducing the temporal distortion of peaks. However, it distorts the timing of the

boundaries, usually making the measurement of a burst duration longer than the actual burst duration. To circumvent the temporal distortion of the burst duration on the averaged EMG signal, some researchers developed a method based on the time at which the averaged EMG signal crosses a selected "threshold amplitude" [Ko88, St86, Ne90, and Clb90].

1.5 Objectives

In animal research, the investigator selects segments of the respiratory muscle signal that contain characteristics of breath patterns. The boundary detection of EMG signals cannot be standardized due to the variation in the background and to the fact that the boundary is also quite variable and its definition is not clear.

The overall aim of this research is to propose an objective method for determining the boundaries for selected segments, implement the method in a computer to develop a segmentation system, and then evaluate the system performance based on boundary detection accuracy in time. The specific goal is to be able to determine transitional segments which include the boundaries and to find the onset and cessation points on two channels of the respiratory EMG signals, e.g. PCA and DEMG. The method will be developed for periods of the burst activity signals which have regular (or normal) respiratory patterns.

The segmentation model will be divided into two major parts: signal segmentation and boundary estimation. In the signal segmentation part, the burst activity will be discriminated from the background with/without tonic activity, resulting in the determination of the transitional segments of variable length. For this operation, multi-channel information involving the airflow signal is utilized to search for points or durations from landmarks in the airflow. In the boundary estimation part, preliminary boundary points (called boundary candidates in this study) of half-wave segments will be detected in the transitional segment and are then estimated by the pattern classification using an artificial neural network. The most probable boundary point (or final boundary) will be

determined based on the boundary estimates.

For boundary detection of the respiratory muscle signals, the segmentation model makes use of a multi-resolution analysis in terms of narrowing down the scope of signal analysis. That is, the EMG signal is segmented, and the transitional segment is further refined into half-wave and finally into point (or sample). Thus, detection of the burst activity supported by multi-channel information can increase the system robustness of the burst activity recognition and the multi-resolution analysis can improve the accuracy of boundary detection in time.

The automated segmentation system which implements the above method will be developed on a NeXT workstation, using an object-oriented programming paradigm. The system will perform the following tasks: preprocessing including bandpass filter and notch filter, signal segmentation to determine the burst activity, transitional, and background segments, boundary detection to find probable boundary candidates, and boundary estimation in which a discriminant function is computed and the candidates are rated by the function. The segmentation system shows graphically the transitional segments, the boundary estimates with probabilities, and the best boundary for the given transitional segment. The evaluation of the system performance will be assessed by comparing the boundaries detected by the system with those determined by EMG experts. The transitional segments are also evaluated.

The proposed method will enhance the recognition rate of the burst activity and the background, and certainly improve the accuracy of boundary points when compared to the conventional method. The automated segmentation system can provide clinicians with a useful tool to understand the respiratory muscle signal and its analysis, and can help them determine the boundaries, thus assessing respiratory patterns.

1.6 Overview

This section presents a brief overview of the remainder of this dissertation. There are three important parts. The first part describes a new boundary segmentation method. The second part is concerned with a software tool of workstation-based environment for digital signal processing algorithm development. It provides users with a means of verifying or evaluating user-developed algorithms by visualizing any kinds of signals and measuring parameters. The second part also describes the implementation of the segmentation model in the NeXT computer. The last part describes the evaluation of the system performance based on the results using a variety of real test signals. Each of these parts is described in a separate chapter.

In this chapter the problem and new approach of the boundary determination of the respiratory EMG signal has been discussed. This chapter has also reviewed previous work in same area and the related research in different fields.

Chapters 2 and 3 present the new model for signal segmentation of respiratory EMG signals. This model consists of four functional steps: noise rejection, burst activity identification and signal segmentation, detection of boundary candidates, and decision making to find boundaries with their probabilities. Chapter 2 deals with the first three parts while chapter 3 does the last part. It also describes a system to derive a discriminant function which will estimate the boundaries.

Chapter 4 discusses two systems. One system describes the interactive design and analysis tool for visualizing and quantifying multiple channels of data. It describes how the tool helps the users in designing and evaluating the user-developed algorithms, such as filters, smoothing algorithms, signal detection, etc. for various signals including EEG, EMG, and milling machine signals. It manages data for both the original signals and the results of the algorithms. The other system describes an implementation of the EMG segmentation model which consists of signal segmentation and boundary estimation. In this chapter, a general system configuration in the form of an object-oriented software paradigm

will be discussed.

Chapter 5 discusses the test and evaluation of the developed segmentation system with selected sets of test data. The first part of this chapter describes the data collection. The second part presents the evaluation results of the system that validate the proposed model for the EMG segmentation problem. A comparison between the results of the system and the visual detection made by two human experts is provided to assess the overall performance of the system.

Chapter 6 summarizes the main idea of this dissertation and suggests directions in which this work can be extended for future studies.

CHAPTER 2

SIGNAL SEGMENTATION METHODOLOGY

The segmentation model consists of four major parts: noise rejection, recognition of burst activity and coarse signal segmentation, detection of boundary candidates, and boundary estimation. This chapter describes the first three parts and the next chapter addresses boundary estimation only. This chapter also presents a general description of the segmentation methodology and the new concept of boundary probability for EMG signals. The new terminologies used in the study are also introduced.

2.1 Introduction

The transition between burst activity and background (or tonic activity) in the respiratory EMG signal is inherently variable. Although a boundary is determined when a threshold is used, it is questionable if the detected boundary represents the true boundary. This implies that it is neither physiologically meaningful nor easy to pinpoint only one boundary point due to the intrinsic fuzziness of the phenomena. Instead of pursuing the boundary point detection, we propose in this work the definition of a “*transitional segment*” which can be defined as a segment of variable duration between the clear burst activity and the clear background.

We propose further to detect, with several criteria, boundary estimates in the transitional segment. To each boundary estimate, a probability will be attributed according to quantitative information obtained from the burst activity segment and the background segment. This probability representation will be called “*boundary probability*” in this study. An example of the transitional segment and the boundary probability is shown in Figure 2.1(a). Basic goals are to detect the transitional segment without error, to determine

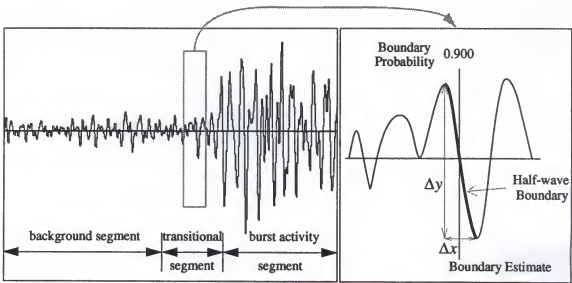
the segment as small as possible, and to correctly estimate boundaries given the burst activity and the background.

The segmentation model is based on a multi-resolution analysis. The conceptual signal flow of the multi-resolution analysis is depicted in Figure 2.1(b). The burst activity of the EMG signal is detected. Thus, segments of the burst activity and the background are roughly discriminated. These two segments are further processed to derive the transitional segment which is a variable length segment where the transition occurs. The transitional segment is again divided into smaller intervals. In this research, two types of transitional segments are defined as follows. The "*coarse transitional segment*" is the interval which is relatively long, but has a high probability of containing the true boundary; the "*fine transitional segment*" is the interval which is narrower than the coarse segment, thus improving the resolution of boundary detection but with a slightly lower probability of not containing the true boundary.

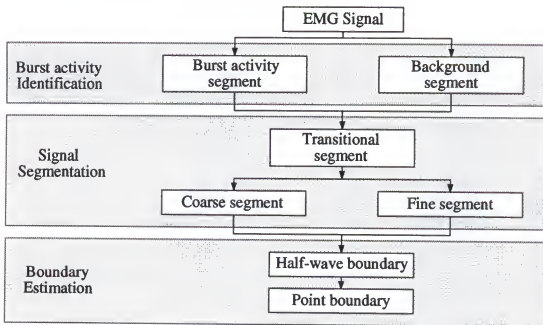
Since the boundaries are located in the transitional segments, the signal analysis is focused in these segments. Before detecting the final boundary point (or sample), the segmentation model deals with half-waves as a basic unit. The half-wave is defined as a line segment between peak-to-trough points which contains the probable boundary point. This half-wave of a variable time-duration and amplitude (Δx and Δy , respectively, in Figure 2.1(a)), is called a "*half-wave boundary*." Now, the resolution of boundary detection is a half-wave. The estimated half-wave boundary is called a "*boundary estimate*." A probability will be assigned to the boundary estimate indicating how close the half-wave boundary is to the true boundary. It will be represented by a vertical bar at the middle point of the half-wave. The graphical description for the above mentioned terminologies is shown in Figure 2.1(a).

Boundary manipulation is further processed to find a final boundary point (or a most probable boundary point). The final boundary, which is a discrete sample in time, is determined in the highest probable boundary estimate.

The multi-resolution analysis narrows the duration of the signal segment targeted for in-depth analysis. As a result the accuracy of the boundary detection is enhanced.



(a) Pictorial description of terminology defined in the study



(b) Description of multi-resolution analysis

Figure 2.1. Description of terminology and multi-resolution analysis

2.2 Overview of Segmentation Model

A general scheme of the automated segmentation model consists of four main phases, which are shown in Figure 2.2: (1) Preprocessing or preconditioning, (2) Signal segmentation, (3) Detection of boundary candidates, and (4) Decision making for boundary probability. The following discussion addresses the overall concept and key points of signal segmentation and boundary estimation methods.

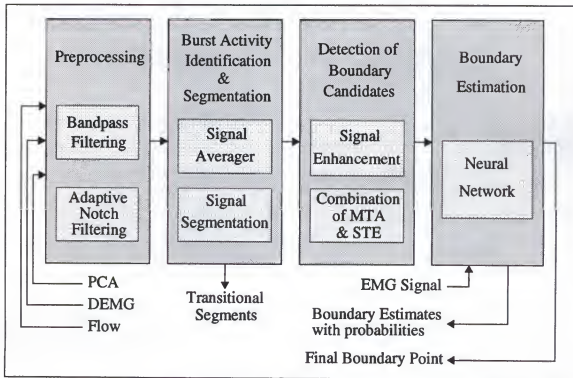


Figure 2.2. Block diagram of the automated segmentation of EMG signals

Almost all the biological signals are contaminated by noise. Before signal processing, undesired signals need to be removed without affecting the characteristics of the burst activity. The three principal noise sources in respiratory EMG signals are the low frequency noise which comes from subject movement and slow activity potentials, the high frequency noise which comes from the recording instrument, and electrical interferences. Thus, the raw signal is first filtered through a bandpass filter to reduce the components of low and high frequency noise. Subsequently, a notch filter using an adaptive noise canceler

is applied, if necessary, to remove the electrical interferences. The sharpness of the notch of the adaptive noise canceler is generally superior to that of a conventional notch filter because the adaptive process maintains the correct phase relationships.

After filtering, burst activity identification and signal segmentation will be performed. The burst activity identification process recognizes the approximate time period during which the burst activities occur with respect to the background segments. The signal segmentation process correctly discriminates the clear segments of the burst activity from the background segments, and then derives transitional segments between them. Therefore, the EMG signal of one breath is divided into three segments.

Since the original EMG signal is a random superimposition of multiple tri-phasic action potentials, the burst activity looks like randomly generated and rapidly changing triangular waves of different amplitude. One of the major difficulties in interpreting EMG signals is that the instantaneous value of the signal contains little information about the overall response of the muscle. For robustness of the burst activity detection, it is better to deal with the global characteristic of the signal obtained by averaging. Averaging several instantaneous responses point-by-point forms a composite waveform representing the mean response. It also reduces the level of random noise in the electrical response of the muscle. In our algorithm, the original EMG signal is transformed into a moving time average signal for segmentation.

The identification of the burst activity is performed on the averaged signal by detecting the slope changes of the high amplitude signal relative to the background level. However, unsteady background and unpredictable burst activity amplitude make the discrimination problematic. Since the mean amplitude of the burst activity and the background segment varies between breaths even in the same subject, the burst activity identification using one channel of signal is difficult. Thus, we propose to rely on multichannel information to recognize the burst activity reliably. In this study, the airflow signal and/or volume signal will be employed. Not only are the breathing patterns of these

signals correlated to the EMG signals, but landmarks such as zero-crossings and extrema are also found easily. These landmarks provide the approximate search periods and reference points to locate more definitively the clear burst activity and the clear background.

The signal segmentation is depicted in Figure 2.1(a) and it is the first step of the multi-resolution analysis. Probable boundary points will be detected only within the transitional segment to avoid detection of wrong boundary points in the background segment which may be caused by the tonic or low frequency activities of high amplitude. The correct determination of the transitional segment is important because all of the following analysis is performed within the transitional segment.

Several points will be considered as probable boundaries inside the transitional segment. Each probable boundary is determined by processing the EMG signal energy with different windows. The signal energy is computed by adding the square of the signal value in a fixed and moving window. The idea of the signal energy processing is to enhance the burst activity with respect to the background within a fixed window size. The window is moved a point at a time, and the output is called an "*energy contour or energy function.*" The energy function is highly sensitive to the signal amplitude change. One difficulty with the energy function is that it is very sensitive to large signal levels. That is, the output of the energy function shows too much fluctuation. A way to control this difficulty is to average the EMG signal first instead of using the raw signal.

As the energy function, we propose a variable short time (or short windowed) signal energy using the moving time average as an input of the energy function. For a fixed window length, we may detect a boundary point. By using different windows, several probable boundaries are detected on which the slope changes of the signal occur. Therefore, several energy contours of different window sizes will be used to determine the half-waves where the probable boundary points reside. These detected probable boundaries (half-waves) are called "*boundary candidates*" in this study. The only requirement for the

boundary candidates is that they should be half-waves close to the true boundary.

Now we will address our proposal for estimating the final boundary. One simple way of improving boundary determination is to average the boundary candidates. However, we seek an unbiased estimate by using the information contained in a collection of representative boundary examples of the transitional EMG signal. We can design a system which can be taught to separate the burst activity from the background through a presentation of useful examples of boundary signals. It is not a simple task to establish a mathematical discriminant for such a obscure problem, so a nonparametric pattern recognition approach is recommended. Therefore, an artificial neural network, a new pattern classification method, is proposed to derive the complex discriminant function for a fuzzy environment by providing quantitative representations of more or less qualitative concepts [Pa89]. The boundary candidates will be rated (or weighted) by the neural network, so that the unbiased boundary estimates are computed.

Another important and quite interesting point in the application of the neural networks is that they can produce the posterior probability estimation. It has been shown recently [El90, Ma91, Li91, and Wh89] that neural nets, when trained using the mean square error criterion with binary desired signals (output of 1 for belonging to a class and 0 otherwise), produce as their output an estimate of the posterior probability of the class given the input. Even though their performance as posterior probability estimators varies slightly as a function of the system error criteria, such as least square error, Kullback-Leibler estimator, and so on, neural nets have been shown to produce approximate posterior probability estimates.

Here we face a question about how to derive the discriminant function through the examples and get unbiased ratings of boundary estimates. The training set for the neural net will include diversified examples where the boundaries of signal patterns are clearly recognizable, so it can acquire the knowledge contained in the training set. That is, the neural net will generalize to a new situation by inducing a relationship that interpolates and/

or extrapolates the characteristics of the training sets. Of course, the neural net will only produce unbiased boundary estimation when the system is trained using suitable network architecture, proper training criteria, and well-defined boundary examples. Quantized raw EMG signal is used as the neural net input to increase boundary detection resolution.

For fuzzy transitional segments, there may be several boundary estimates with different probabilities. The final boundary point (or sample) for a given transitional segment will be determined based on boundary estimates with the highest probability. The basic criterion is to detect the point crossing the background level.

The following sections describe the first three blocks of Figure 2.2, that is, preprocessing, burst activity identification and segmentation, and detection of boundary candidates.

2.3 Preprocessing

The basic concept of preprocessing is to reduce the effect of noise, resulting in a higher signal-to-noise ratio. The signal conditioning will remove unwanted components without severely altering the important information of the signal of interest. Thus, this section is concerned with the preconditioning of the raw signals, leading to rejection of noise sources such as electrodes and their displacement, subject movement, test equipment, and electrical power line.

2.3.1 Noise Sources of Respiratory EMG Signals

From a physical point of view, the respiratory EMG signal, even if carefully and cleanly recorded, is always composed of both desired signals, which are the muscle's action potentials to be detected, and a certain amount of the background or noise. The noise can be categorized into three parts.

Biological or bioelectrical noise. Although any biological source of electrical potentials in the body of the experimental subject is actually a source of cross-talk rather

than noise, some sources are so unlike the EMG sources that they are considered as noise. This category includes electrocardiograph and respiratory waves as well as neural evoked potentials. The slow potential fluctuations accompanying the expansion and contraction of the chest wall are actually forms of motion artifact. Slow-motion artifacts can usually be filtered to a negligible size. However, if the method of analysis includes rectification and integration, even small residual, low-frequency signals may generate large outputs because of their large area under the curve. If the study is one of cyclic events such as activity of the respiratory muscles, both EMG signals and movement will be time locked to the respiratory cycle; thus the biasing effect, even of small artifacts, may be quite significant. Motion artifacts can be a particularly difficult source of randomly occurring, relative large-amplitude, low-frequency waves. Another noise source of this type comes from MUAPs firing at a distance from the recording needle tips, which is colored noise corresponding to low amplitude, low frequency signals. This low frequency noise can be removed by highpass filtering.

Electrical interferences. The most common problem in bioelectrical recordings is the presence of interference from electrical power sources, i.e., 60 Hz and its harmonics. It may be caused by poor (high resistance) contact between electrodes and muscle (or skin), inadequate shielding, inadequate grounding or common connections on the subject, broken or high resistance cables, or by poor common mode rejection in the amplifier. If this resistance changes, the amplitude of the recorded signal may also change. This type of interference can be eliminated by using one or several notch filters.

Instrumental noise. Instrumental noise includes all the sources of noise that are part of the physics of electrons and electronic components, consequently forming an integral part of any recording process such as the impedance of electrodes, the design and construction of preamplifiers and amplification system, and cables. This type of noise can generally be considered random, very wideband noise of a Gaussian distribution that lacks

an apparent pattern or predominant shape. Of course, this means that when narrow band filtering is used, signatures quite like real EMG potentials will occasionally appear. Most such noise effects can be evaluated by simply examining the output generated in the normal recording configuration at a time when no signal is expected.

In a muscle signal with weak contraction, the measured signal is usually made up of two types of intervals: (1) intervals of background, which contain only noise (instrumental noise with or without biological perturbations) and are free of desired signals. (2) intervals of muscle activation signal which contain desired signals (elementary or complex superimposed MUAPs) disturbed by instrumental and biological noise. If different characteristics (amplitude, frequency or power spectra) are assumed for each of these two types of intervals, a relatively simple method (level detector for amplitude characteristics, digital filter for frequency characteristics) can be developed which is capable of producing a segmentation of the signal. Also, in each activation interval, evaluation of the amount of disturbing superimposed noise will make it possible to detect the desired signals (MUAP or sum of MUAPs). However, in the case of respiratory EMG signals with strong contraction, the background does not consist only of noise. It shows weak characteristics of the burst activity since the associated muscle is not completely relaxed in the background interval, producing the tonic activity. Thus, the discrimination of the burst activity from the background becomes unclear.

2.3.2 Noise Rejection

There are two EMG signals to be processed: PCA (posterior cricoarytenoid) and diaphragm muscle signal (DEMG). The spectrum of the respiratory muscle signals show that there is no noticeable discrimination between the burst activity and the background, and no sharp peak frequencies.

To reduce low frequency components which are mainly generated by the biological noise and slow-time MUAP waveforms, and to reject high frequency components which

are generated by the instrumental noise, a FIR (Finite Impulse Response) bandpass filter was applied. The impulse response of the bandpass filter was computed by an optimal design technique using Parks and McClellan algorithm [Par72]. One should be careful in designing the low cutoff frequency of the bandpass filter since the basic criterion to discriminate the signal is amplitude or energy comparison and the low frequency noise of relatively high amplitude directly affects the signal amplitude. This type of low frequency signal significantly affects the determination of boundary points, i.e., the measured activity period may be longer than the actual period of activity. However, it is difficult to clearly delineate the low frequency limits of the bandpass filter because the low-frequency band signals are always composed of parts of the burst activity signal and noise components. In other words, a filter of sharp cutoff frequency with narrow transitional bandwidth is necessary to increase the signal-to-noise ratio by significantly reducing the effect of low frequency noise. The bandpass filter is very effective in keeping the major frequency components of the burst activity despite a probable slight reduction of the amount of the energy of the burst activity.

After bandpass filtering, another step in the preprocessing stage is to reject the electrical interferences. This noise source is frequent in the diaphragm muscle signal and unusual in the PCA signal. An adaptive noise canceler (ANC) will be implemented instead of using a conventional notch filter [Hu73, Gl77, and Sh80]. Strictly speaking, it is not adaptive since the reference input is a fixed frequency signal. However, it uses an adaptive filter. We found out that these noise sources occur with an irregular number of harmonics with different amplitude. A fixed number of ANC is applied in parallel to reject these harmonics.

The ANC filtering is skipped when the signal is not contaminated by electrical interferences. Figure 2.3 shows the filtering procedure of the respiratory muscle signals.

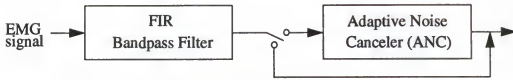


Figure 2.3. Flow of signal preprocessing

2.3.3 Design of Bandpass Filter

The frequency distribution of the recorded EMG signal ranges from around 25Hz to 500Hz. However, these two muscle signals have more energy in the band of 60 - 450Hz which is shown in Figure 2.4(a). Determination of the high-frequency transitional band is so simple that the frequency band of 450 to 550Hz was selected. Determination of the low-frequency band was made based on experimentation. There are two type of signals, shown in Figure 2.5, to be considered for determining the cutoff frequencies of the low-frequency transitional band. Segment *A* in figure (a) shows an example of the signal to be rejected, which is a typical motion artifact, while segment *B* of figure (b) shows the signal to be kept. The major decision criterion utilized consists of visually identifying the amount of reduction of low frequency components. Thus, appropriate segments of the EMG signal, especially at the end of burst activity were collected and the power spectrum computed. The low frequency components (0 to 60Hz) were corrupted with noise components as the comparison of spectra of EMG signals and collected segments shows (refer to Figure 2.4(a) and (b)).

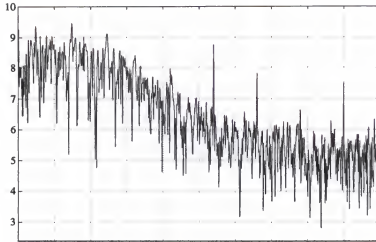
To reduce the low frequency noise as much as possible, a high-order bandpass filter was applied to increase stopband attenuation and decrease transitional width of the low frequency band. Therefore, the 63rd order FIR bandpass filter with a cutoff frequency of 58Hz-128Hz and 450-550Hz transition bands was applied. Figure 2.4(c) shows the frequency response after filtering. The bandpass filter was designed in the Monarch software [Mo88]. In this package, the optimal design technique using the Remez algorithm,

which is one of the FIR design techniques, was implemented. The criterion of optimality is the minimax norm, given by

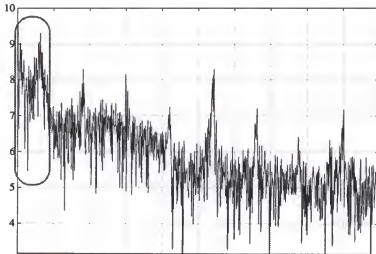
$$e(\omega) = |E_N(e^{j\omega})| = \min [\max (E_N(e^{j\omega}))]$$

where, $E_N(e^{j\omega}) = H(e^{j\omega}) - H_N(e^{j\omega})$ for order N , $H_N(e^{j\omega})$ represents the transfer function of the realized filter and $H(e^{j\omega})$ is the designed transfer function which has minimum and maximum values.

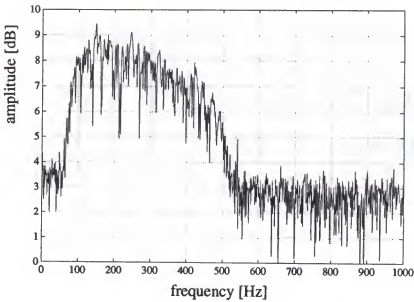
Figure 2.6 shows impulse, frequency, and phase responses of the 63rd order FIR filter with 0.014157 deviation and 2.19121 maximum gain in a passband, and 50dB of out-of-band attenuation. Figure 2.7 shows an example of the bandpass filtering for PCA and DEMG signals.



(a)
Frequency response
of EMG signal

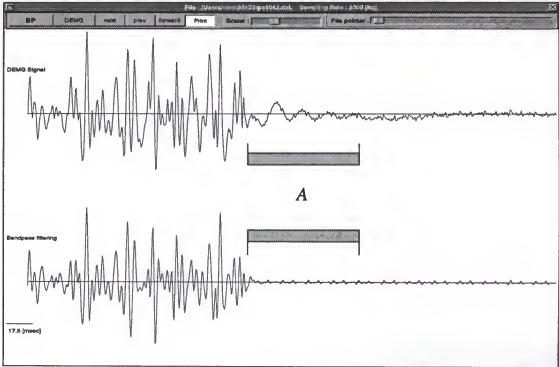


(b)
Frequency response
of motion artifact
background

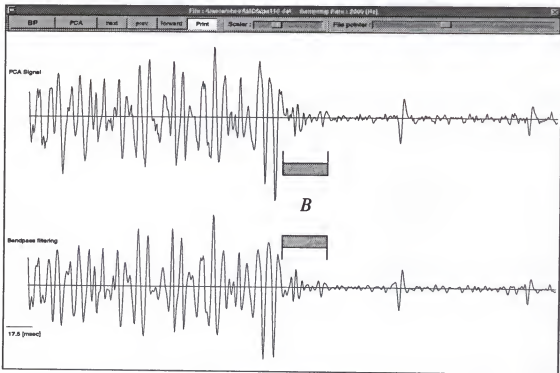


(c)
Frequency response
of bandpass-filtered
signal

Figure 2.4. Example of frequency response of EMG signal

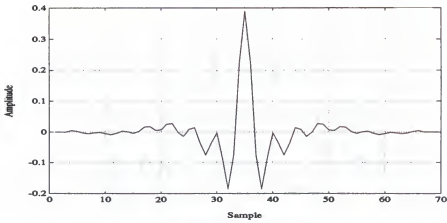


(a) Example of signal to be rejected

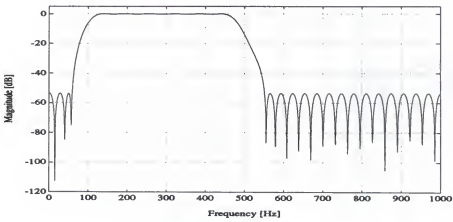


(b) Example of signal to be kept

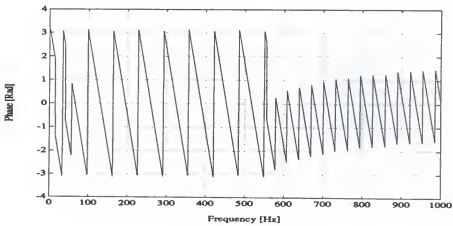
Figure 2.5. Signals used for decision of low cutoff frequency



(a) Impulse response



(b) Frequency response



(c) Phase response

Figure 2.6. Characteristic of bandpass filter

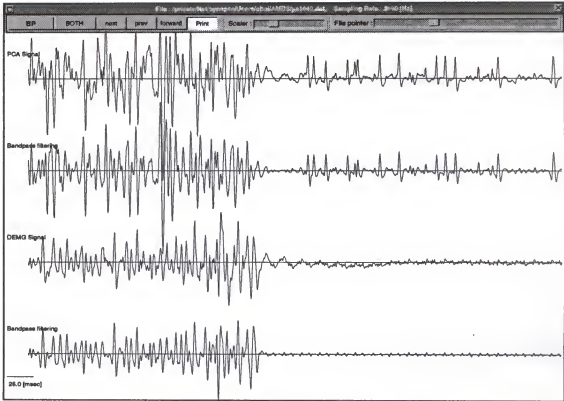


Figure 2.7. Example of bandpass filtering

2.3.4 Design of an Adaptive Noise Canceler (ANC)

In this section we are concerned with the rejection of electrical harmonics. The block diagram of the ANC is shown in Figure 2.8. The primary input, d_k , consists of the signal plus noise. The reference input, x_{1k} and x_{2k} , is not correlated with the signal but, correlated in some unknown way with the primary noise. The reference is filtered to match the primary signal's noise and then subtracted from the primary. The error signal, ϵ_k , to the adaption algorithm is therefore the output of the ANC system. The adaption is accomplished by feeding the system output back to the adaptive filter and adjusting the filter coefficients, w_{1k} and w_{2k} , through an adaptive algorithm (LMS box in figure) to minimize the total system output power. In an adaptive noise canceling system, the system output serves as the error signal for the adaptive process. The adaptive filter used to perform

the noise canceling is a transversal filter. The input sequence sampled at a rate $1/T$ is applied to an N -stage tapped delay line (TDL). The values at the N taps of the TDL at time k constitute the elements of the reference N -vector x_k . The outputs of the taps are weighted and summed to give the adaptive filter output $y_k = w_k^T x_k$.

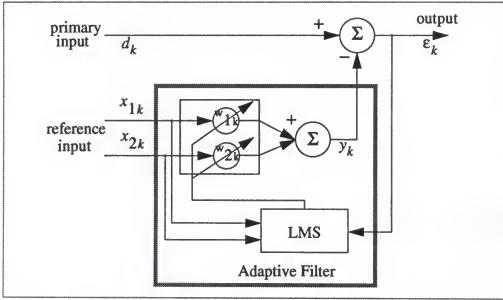


Figure 2.8. Block diagram of the adaptive noise canceler

The adaptation algorithm most often used to set the weights of the filter is the least-mean-square (LMS) algorithm [Wi66] given by the equations

$$y_k = w_k^T \cdot x_k$$

$$\epsilon_k = d_k - y_k$$

$$w_{k+1} = w_k + \alpha \epsilon_k x_k$$

LMS is an iterative gradient-descent algorithm that uses an estimate of the gradient on the mean-square error surface to seek the optimum weight vector at the minimum mean-square error point. The term $\epsilon_k x_k$ represents the estimate of the negative gradient, and the adaptation constant α determines the step size taken at each iteration along that estimated negative-gradient direction. The true negative gradient is given by the expected value of $\epsilon_k x_k$. If α is chosen properly, such that small steps are taken, adaptation noise due to the

error in the gradient estimate is averaged out.

With undesired sinusoidal noise such as an electrical interference, the suggested approach is to select a different set of inputs and outputs as illustrated in Figure 2.8 which shows a single-frequency noise canceler with two adaptive weights [Wi85]. The reference input is a pure cosine wave, $C \cos(\Omega_0 t + \theta)$. The reference input is sampled directly, giving x_{1k} , and after undergoing a 90° phase shift, giving x_{2k} . The transfer function can be obtained by analyzing signal propagation from the primary input to the system output. The procedure for updating the weights is given by

$$w_{1,k+1} = w_{1k} + 2\mu\epsilon_k x_{1k}$$

$$w_{2,k+1} = w_{2k} + 2\mu\epsilon_k x_{2k}$$

where μ is a constant that governs stability and rate of convergence.

If we let $w_0 = 2\pi f_0 T$ sampled at intervals of T seconds, the sampled reference inputs are

$$x_{1k} = C \cos(kw_0 + \theta)$$

$$x_{2k} = C \sin(kw_0 + \theta)$$

The 3-dB bandwidth (BW) is shown to be

$$BW = 2\mu C^2 \text{ rad} = \frac{\mu C^2}{\pi T} \text{ Hz}$$

Electrical interferences are 60Hz and its harmonics. Signal amplitude of 60Hz and higher harmonics of greater than 480 Hz was significantly reduced in bandpass filtering. Thus, we are interested in 2nd harmonic(120Hz) through 8th harmonic(480Hz). Through experimentation, it is verified that major harmonics of interest are 180, 300, and 420Hz. The formation of multiple notches has been achieved by using an adaptive filter with multiple weights. Two weights are required for each sinusoid to achieve the necessary filter gain and phase. The system block diagram of multiple noise canceler is shown in Figure 2.9.

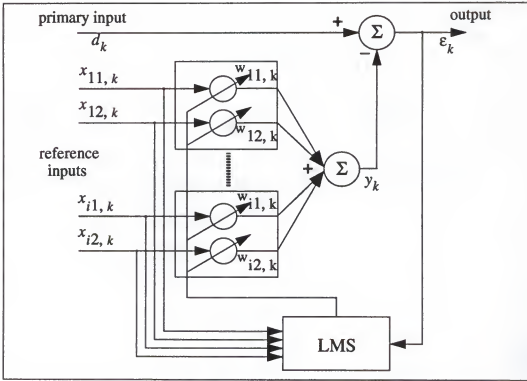


Figure 2.9. Adaptive noise canceler for multiple interferences

In this figure,

$$x_{i1,k} = C \cos(2\pi(f_i/f_s))$$

$$x_{i2,k} = C \sin(2\pi(f_i/f_s))$$

where f_i is a frequency to be rejected, f_s is the sampling frequency of 2000Hz, and $C=1$.

$$w_{i1,k+1} = w_{i1,k} + 2\mu\epsilon_k x_{i1,k}$$

$$w_{i2,k+1} = w_{i2,k} + 2\mu\epsilon_k x_{i2,k}$$

$$y_k = \sum_{i=1}^n (x_{i1,k} w_{i1,k} + x_{i2,k} w_{i2,k})$$

where n is the number of multiple sinusoidal reference inputs, i.e., $n = 3$ corresponds 180, 300, and 420 Hz.

To test the frequency response of the system, a sinewave sweep was used as input to the system. The frequency of the input was varied from 1 to 1000Hz. The system responses for an experimental sweep frequency are shown in Figure 2.10. Figure 2.10(a)

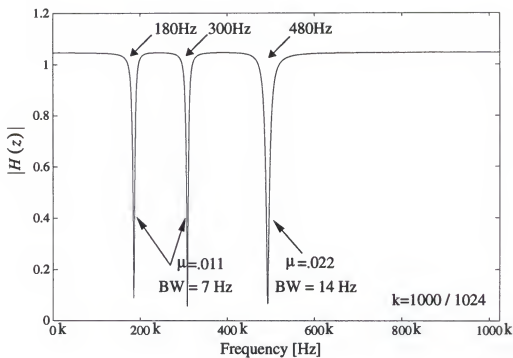
shows the frequency response which is calculated by $|\text{FFT}(\epsilon_k)/\text{FFT}(d_k)|$. As indicated on the figure, the bandwidth of two lower frequencies (BW=7Hz at 180 and 300Hz) is half that of the last one (BW=14Hz at 480Hz). The μ value is 0.011 for two lower frequencies and 0.022 for the highest frequency. If μ is reduced, the response of the system better matches the response of an ideal notch filter. The phase response of the system is plotted on Figure 2.10(b). The phase is distorted at the notch frequency, and remains at zero during the rest of the spectrum. The phase response figure shows that the ANC is not linear phase filter. The filtered signal does not follow the original signal shape.

However, considering that the recorded EMG burst activity is the summation of mutually interfered MUAPs, the linear phase of the filter characteristic is not crucial in boundary detection. It is rather important to increase the energy of the burst activity while reducing the background. Therefore, it is highly recommended to utilize the adaptive noise canceler.

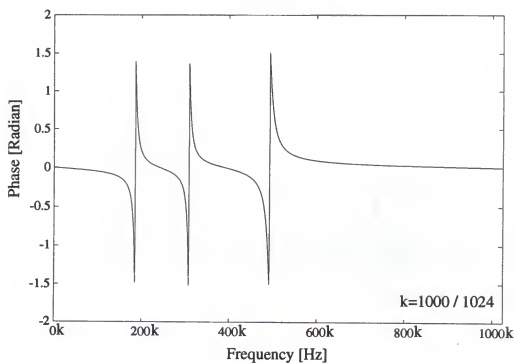
Figures 2.11 and 2.12 show examples of ANC filtering with emphasizing adaptation effect. The (b) in both figures is the ANC result of a notch frequency of 180Hz and (c) is the result of notch frequencies of 180Hz and 300Hz using same μ value of 0.011. The signal level of the background is significantly reduced. The quantitative comparison of background amplitude is shown in Table 2.1 by computing the average of absolute value for 600 samples. When the ANC is processed at different notch frequencies, the mean value of the background is roughly reduced by half.

Tabel 2.1. Average value of the background segment in Figure 2.11 and 2.12

	Figure 2.11	Figure 2.12
(a) original signal	17.1281	17.1581
(b) ANC output for 180Hz	8.7765	8.8901
(c) ANC output for 180 & 300Hz	4.6970	4.5508



(a) Frequency response



(b) Phase response

Figure 2.10. Responses of ANC for experimental sweep frequency

The adaptation time of ANC algorithm inherently depends upon the μ value: the larger the μ , the more rapid the convergence. The signal shape of the filtered output will also vary according to the μ value due to the different speed of weight convergence. Generally, the ANC algorithm at current time point also relies on the previous value of data. In particular, an interesting point in adaptation is that it depends upon the change rate of signal energy. Figures 2.11 and 2.12 show ANC results for two signal transitions, i.e., from the burst activity to the background and vice versa. Both figures illustrate the effect of different adaptation time according to the change in energy (i.e. change from large energy of the burst activity to small energy of the background or vice versa, refer to circles A and B in both figures). The original signal of Figure 2.12(a) is same as Figure 2.11(a) but it is simply reversed in time.

Different adaptation times are caused by the way the energy change (decrease or increase), which is denoted as C and D in (a) of both figures. The weight change, Δw , in the adaptive filter is proportional to the error, ϵ_k , i.e. $\Delta w = w_{k+1} - w_k = 2\mu\epsilon_k x_k$. We will assume that the filter is adapted prior to the signal change. In Figure 2.12(a), when the signal changes rapidly, the ϵ_k value becomes large, resulting in a rapid change of Δw , that is, fast adaptation. Thus, signals of the D region quickly follow the original signal. However, the ϵ_k of the C region in Figure 2.11(a) is relatively large compared to that of the Figure 2.12(a). When the signal amplitude decreases, so does the error ϵ_k , that is, the change rate of Δw becomes small. Thus, the D region of Figure 2.11(a) adapts slower. This characteristic is not present in fixed coefficient linear filters, but does exist in linear time varying filters.

The weights change rate for Figures 2.11 and 2.12 is shown in Figure 2.13, in which the weight vectors associated with the notch frequencies are displayed. They are plotted versus the iteration for comparison purposes with the original signal. In all four plots there exists a significant change in weight adaptation around iteration 400 and 1300. These changes can be attributed to the fact that the original signal in Figures 2.11 and 2.12 shows

a significant change in wave shape at these samples, as explained in above paragraph. Thus, nonstationarity causes fluctuations in the weight convergence. It is interesting to compare the weights change rate before rapid signal change (denoted E in figure) between (a) & (c) and (b) & (d) in Figure 2.13. These figures demonstrate the adaptation difference between A in Figure 2.11 and B in Figure 2.12.

It has been shown that B in Figures 2.11 and 2.12, both filtered signals provides a clearer boundary when the algorithm is processed from the background to the burst activity. However, in the signal change from large energy to small energy, which is denoted by A , the boundary after filtering is not always clear. In terms of boundary discrimination, we need to avoid this type of phenomenon. The following paragraph explains, through example (Figure 2.14), how to deal with this problem for more precise boundary detection.

The first signal in Figure 2.14 is the original signal. The second is the output of ANC processing with a forward direction of data stream. The third is the ANC output after processing with reverse direction under the same conditions. Dependent upon the direction of data stream to the algorithm input, Figure 2.14 shows their differences, especially around boundaries. This means that for cessation boundary detection the data flow direction must be reversed. Therefore, the integration of these two filtered signals can provide precise determination of both boundaries which is the fourth signal in the figure. Since the test EMG signal is a fixed length of data segment, the data buffer of the EMG signal is first provided by the ANC algorithm for forward processing and the result is saved in a file. The same algorithm is repeated for reverse processing by providing the data buffer reversely, and the result is saved into another file. A combining strategy is to detect the middle points of the burst activity and the background, to choose the onset and cessation segments which are shown in the figure, and then to put together the selected segments. The middle points are calculated from zero-crossings of the Flow signal. Figure 2.14 graphically illustrates the combining strategy. This is possible since we are only interested in the boundary determination.

To find out the convergence of the algorithm, the stationary segment of around 450 sample of Figure 2.11(a), which is a part of the background, is chosen. Figure 2.15 shows the weight tracking on the performance surface contour in the selected segment. Figure 2.16 shows a learning curve or adaptation speed with $\mu=0.011$. The learning curve is characterized by [Wi85]

$$\xi - \xi_{min} = (W - W^*)^T R (W - W^*)$$

where ξ is mean-square error (*MSE*) and ξ_{min} is minimum *MSE*. The $\xi - \xi_{min}$ is a quadratic form in a weight vector, W , which reaches its minimum value when W equals an optimal weight vector W^* . An input correlation matrix, R , means that each optimum weight is independent from other weights. The theoretical values of the time constant of adaptation is

$$T_{mse} = 1 / (4 \cdot \mu \cdot \lambda)$$

where, λ is an eigenvalue of input correlation matrix. With $\mu=0.011$ and $\lambda = 0.5 \cdot C^2=0.5$ ($C=1$), the theoretical time constant is $T_{mse}=45.4545$. From the learning curve of Figure 2.16, the slope is estimated to be approximately one decade in 100 iterations. Since a decade is a factor 10 or $e^{2.3}$, T_{mse} is observed to be the order of $100/2.3$ or 43.4783 iterations. Thus, the theoretical and measured value are same in the order of accuracy.

Misadjustment or “rattling” measurement, M , provides a measure of the difference between actual and optimal performance averaged over time. The theoretical misadjustment is given by [Wi85]

$$M=\mu \cdot \text{tr}[R]=0.011, \text{ where } \text{tr}[R]=\sum_{n=1}^L \lambda_n=1$$

The equation to calculate the misadjustment is

$$M=(\text{excess } MSE) / \xi_{min}$$

where (excess *MSE*) means average *MSE*. This measure should be applied to a stationary segment after adaptation. From the learning curve it is expected that after 250 iterations, the weights are adapted. Thus, the weights between 250 and 450 point are applied to compute measured misadjustment in terms of time average, resulting in 0.008114.

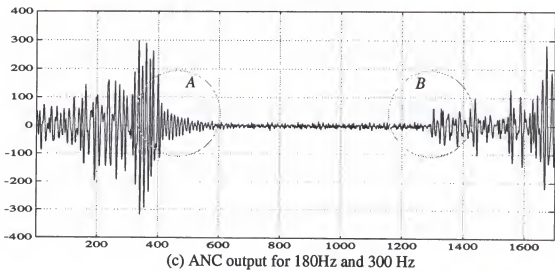
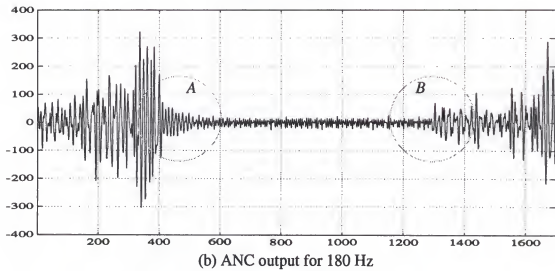
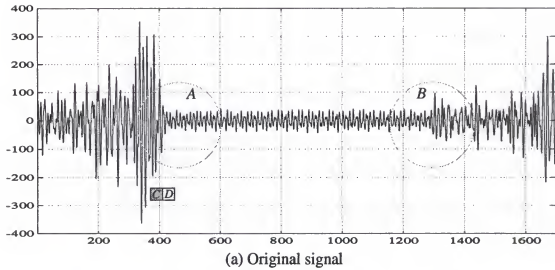


Figure 2.11. ANC filtering and adaptation effect for change of the signal energy - #1

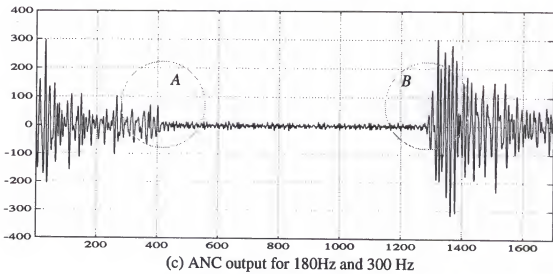
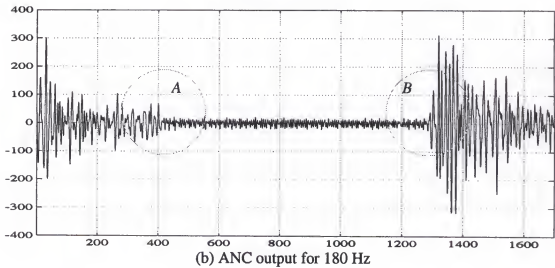
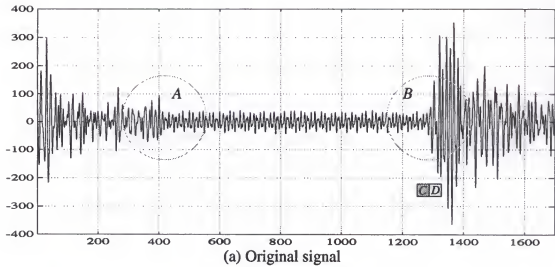
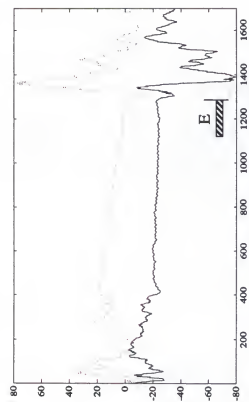
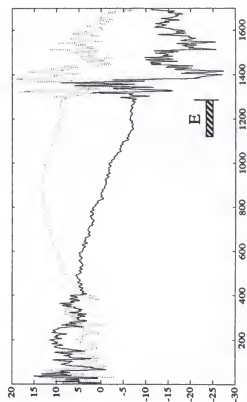


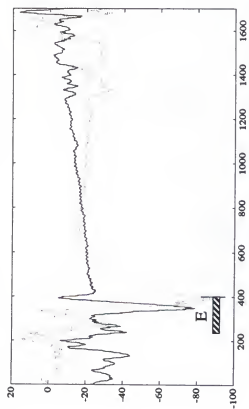
Figure 2.12. ANC filtering and adaptation effect for change of the signal energy - #2



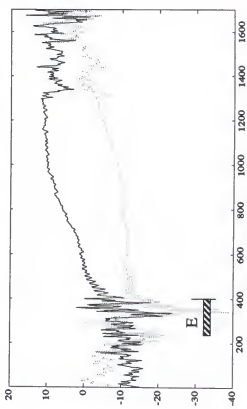
(c) Weights adaptation in Figure 2.12(b)



(d) Weights adaptation in Figure 2.12(c)



(a) Weights adaptation in Figure 2.11(b)



(b) Weights adaptation in Figure 2.11(c)

Figure 2.13. Weights adaptation

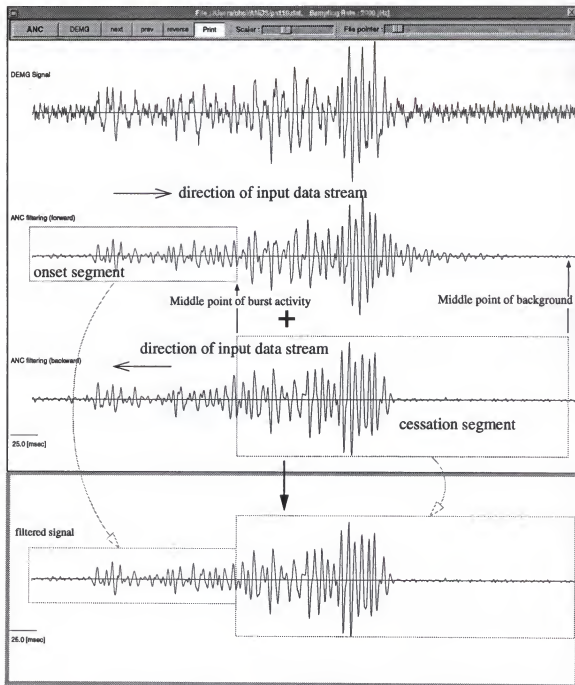


Figure 2.14. ANC outputs for different direction of input data stream and its combination

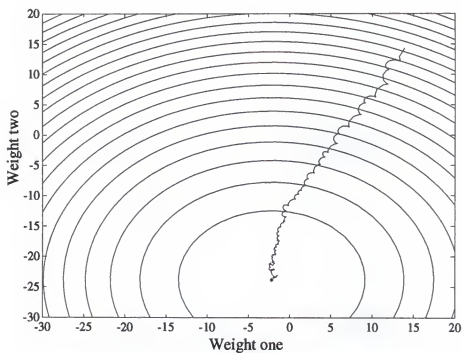


Figure 2.15. Contour and weight value tracking with $\mu = 0.011$

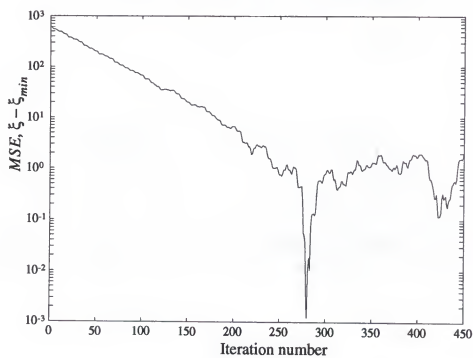


Figure 2.16. Learning curve with $\mu = 0.011$

2.4 Burst Activity Identification and Signal Segmentation

In this phase, the burst activity is recognized from the background, resulting in the determination of a transitional segment in which the boundary points are located. All signal analyses are performed on the filtered signals.

2.4.1 Burst Activity Identification

Burst Activity identification is comprised of detecting large slope changes of the average signal amplitude with respect to the background level. We decided to identify the burst activity using multichannel information. The airflow or its integral, the volume signal, is tightly related to the respiratory pattern. However, the Flow signal is more stable than the EMG signals, which makes it easier to analyze. Some landmarks, i.e., zero-crossings or peaks, of the Flow signal provide pointers to find the maximum burst activity and the minimum background in one cycle of breath. Further, using the statistical information of the minimum background, we can detect coarse points where the burst activity starts and ends. The coarse points lead to a discrimination of the burst activity from the background.

2.4.1.1 Signal averager - Moving Time Average (MTA)

The use of an averaged signal derived from the raw EMG signal provides a clearer indication of the burst activity force. For discrete-time sequences, a common smoothing operation is a moving time average, where the smoothed value, $y(n)$, for any time sequence n , is an average of values of input signal, $x(n)$, in the vicinity of n . The basic idea is that by averaging values locally, rapid variations from point to point will be averaged out and slow variations will be retained, corresponding to lowpass filtering. As a generalization of the moving time average filter, we can consider averaging over $N+M+1$ neighboring points, that is, to use a difference equation of the form

$$y(n) = \frac{1}{N+M+1} \sum_{k=-N}^M x(n-k)$$

Here the corresponding impulse response is a rectangular pulse. A delay of N points will

make these FIR filters causal. The filter frequency response is

$$Y(Z) = \frac{1}{N+M+1} \sum_{k=-N}^M X(Z) Z^k$$

$$H(w) = \left. \frac{Y(Z)}{X(Z)} \right|_{Z=e^{jw}} = \frac{1}{N+M+1} \sum_{k=-N}^M e^{-jwk}$$

or

$$H(w) = \frac{1}{N+M+1} e^{(-jw(N+M))/2} \cdot \frac{\sin[(N+M+1)w/2]}{\sin(w/2)}$$

where w is digital frequency. The cutoff frequency of the MTA filter corresponds approximately to $[(2 \cdot \pi) / (N+M+1)]$, i.e., $(f_s = \text{sampling frequency}) / (N+M+1)$.

A further generalization of the MTA filter can be made by forming a weighted average of $(N+M+1)$ neighboring points, that is, by using a difference equation of the form

$$y(n) = \sum_{k=-N}^M b_k x(n-k)$$

where the coefficients, b_k , can be selected to achieve the prescribed filter characteristics. The above equation is then in the form of a general nonrecursive difference equation. It is clear that by an iterative procedure for the optimum weighting coefficients such as Parks and McClellan method, the transition band can be sharpened.

The window length of MTA filters currently used in processing the respiratory muscle signal is 81 ($N=M=40$) and 21 ($N=M=10$) for which the determination will be explained in the following section. For 21 neighboring points,

$$y(n) = \frac{1}{21} \sum_{k=0}^{20} x(n-k)$$

$$H(Z) = \frac{Y(Z)}{X(Z)} = \frac{1}{21} \sum_{k=0}^{20} Z^{-k} = \frac{1}{21} \cdot \frac{1-Z^{-21}}{1-Z^{-1}}$$

The system response corresponds to sampling the Z -transform $H(Z)$ at 21 points equally spaced around the unit circle, with pole-zero cancellation at $Z=1$. The cutoff frequency of this lowpass filter is $(2\pi)/N$, i.e., $(f_s = 2000) / 21 = 95.24\text{Hz}$. For 81 window length, the cutoff frequency is 24.70Hz.

Windowing of MTA. When the MTA is defined a rectangular window is assumed, i.e.,

$$y(n) = \frac{1}{K} \sum_{i=-\infty}^{\infty} w(n) x(n-i) = \frac{1}{K} \sum_{i=-N}^N x(n-i)$$

$$\text{where, } w(n) = \begin{cases} 1, & -N \leq n \leq N \\ 0, & \text{otherwise} \end{cases}$$

The selection of window type does not affect much the smoothing of the burst activity envelope. An example of several windowing functions of length 51 is shown in Figure 2.17. The first signal in the figure is an original EMG signal, the second is an output of the rectangular window, the third is for Bartlett window, the fourth is for Hanning window, the fifth is for Hamming window, and the sixth is for Blackman window. In this research, the Hanning window is used because it provides a little better smoothing effect for the burst activity envelope. The Hanning window is specified by the following equation:

$$w(n) = \frac{1}{2} \left[1 - \cos \left(\frac{2\pi n}{N-1} \right) \right] \text{ where, } 0 \leq n \leq N-1$$

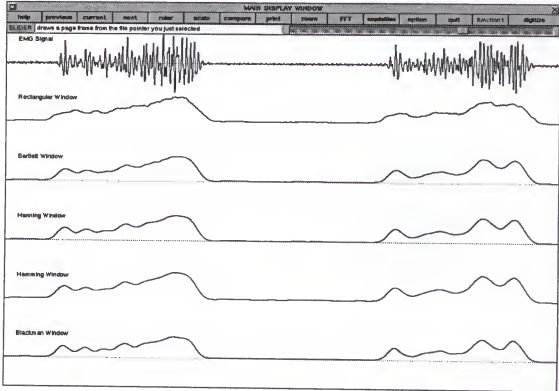


Figure 2.17. Smoothing effect of several different windows

What is more important is the determination of the window length. By increasing the window length, N , we can get a much smoother envelope. For a window length which is too small, the MTA output shows fluctuations for smaller amplitudes of the signal. Generally speaking, a large window length is more appropriate than a smaller one in terms of the burst activity recognition because it shows the global representation of the burst activity energy. However, it is hard to fix the window length due to the diverse patterns at the boundary. The problem in the determination of the window length is false detection of the burst activity threshold according to the searching direction, which is exemplified in Figure 2.18 and 2.19.

For short window length, for example $N=21$ in Figure 2.18(b), the detection of the threshold will be A , which is a correct detection, for the search of the forward direction, and B , which is wrong detection, for backward direction. Thus, a short window MTA with backward searching may cause the wrong detection of the threshold for the transitional signal of the Figure 2.18(a). But, we can find a different answer for the transitional signal of Figure 2.19(a) although the same window length of 21 is used. The threshold detection of Figure 2.19(b) is E , which is wrong detection, for forward direction and F , which is correct detection, for backward direction. For large window length, for example $N=91$ in Figure 2.18(c), the detection of the threshold will be C , which is the same point for both forward and backward direction, and it shows correct detection. However, for MTA of too large window length, for example $N=111$ in Figure 2.19(c), the threshold detection will be G for both searching directions which shows the wrong detection. It includes MUAPs near a cluster of the burst activity as a part of real burst activity. In other words, sparse burst activity in the background may be recognized as the real burst activity.

The figures demonstrate that the window length and searching direction will affect the detection of the burst activity. Empirically, the window length of $N=81$ was verified to be appropriate for the identification of the burst activity, with forward searching for the cessation threshold and backward searching for the onset threshold.

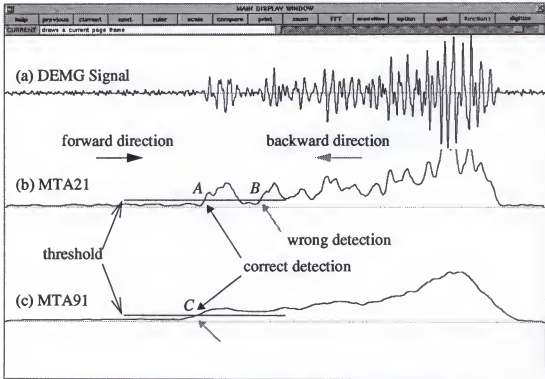


Figure 2.18. Threshold detection with different window length and different searching direction - example 1

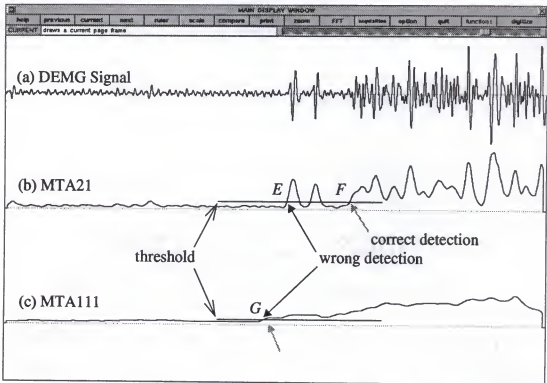


Figure 2.19. Threshold detection with different window length and different searching direction - example 2

2.4.1.2 Relationship between flow and EMG signals

The burst activity on the MTA signal is recognized based on multichannel information. Thus, the relationship between the Flow signal and EMG signals will be reviewed before discussing the identification of the burst activity and signal segmentation.

Figure 2.20 shows the normal relationship between the Flow signal and two EMG signals, PCA and DEMG. An interval of $F2$ and $F3$ in the figure is inspiration and an interval of $F3$ and $nF2$ is expiration. The points, $nF1$, $nF2$, and $nF3$, mean zero-crossings for the next breath. A period of breath is the sum of inspiration and expiration. The interval, A , indicates an interval where an onset point exists while B indicates an interval where a cessation point exists. We want to determine reference points from the Flow signal to recognize the burst activity. Our method is the following.

First, all zero-crossings such as $F1$, $F2$, and $F3$, are detected. The middle point, $min_x = (F3+F2)/2$, is the point which indicates the end of the burst activity. Another middle point, $C = (nF1+F3)/2$, is the point which is always in the clear background. The starting point of inspiration, $max_x = F2$ for current breath (or $nF2$ for next breath), is the point which indicates for the DEMG signal the starting part of the burst activity.

As can be seen in the figure, the PCA signal shows two clusters of the burst activity in a breath. The burst activity between them sometimes is small, at the same level of the background. Since the onset is somewhere in the A interval on the first cluster, we need to locate in the PCA signal the middle point, T , of the first cluster of the burst activity which corresponds to max_x of the DEMG signal. The point T is detected as the MTA maximum by searching from $(nF1-50 \text{ msec})$ to K , where K is determined as the MTA minimum within the interval $(nF1+(nF2-nF1)/4)$ to $nF2$.

Thus, the recognition of the burst activity is performed on the MTA signal based on the search period of $S1$, $S2$, and $S3$.

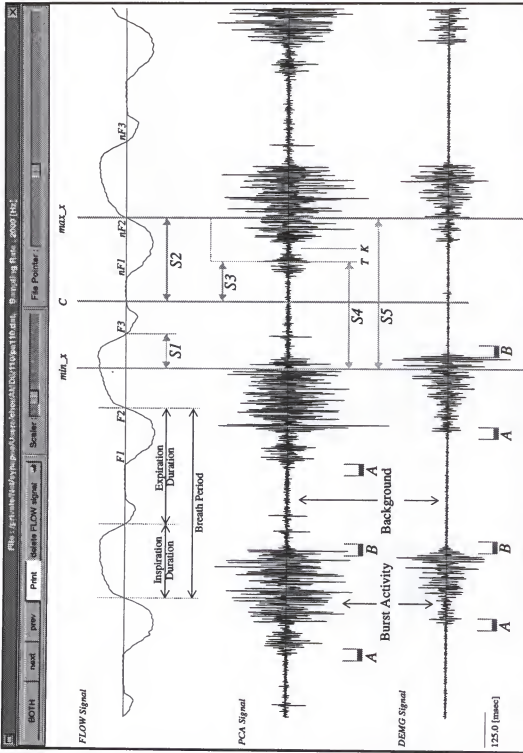


Figure 2.20. Relationship between Flow and EMG (PCA and DEMG) signals

Recognition of the burst activity. To detect the signal slope of the averaged signal(MTA81) for recognition of the burst activity, a reference value, which is the amplitude level of the background, is necessary. This value does not change rapidly although it is not always the same among breaths. Thus, we propose an adaptive normalization to track the level change of the mean background activity. According to our experimentation and other research [Br84, Ya80, Ne90, and Cl90a], the following procedure has proven useful to set a threshold of the background level for recognition of the burst activity.

The “*minimum-valued segment*, ” of approximately 100 msec duration in the background is determined by searching from min_x to max_x ($S4$ in Figure 2.20 for the PCA and $S5$ for the DEMG). The mean amplitude value of the minimum-valued segment is calculated. The burst activity is recognized when the signal amplitude exceeds or returns to a threshold value defined as $threshold = k \cdot mean$. Figure 2.21 explains graphically the recognition of the burst activity for the DEMG signal, and a similar processing is used for the PCA signal.

By searching from $F3$ to min_x point ($S1$ on the MTA81 signal), we detect a point, P_{ic} , by using an appropriate estimate of k . And by searching again from C to max_x in the MTA81, another point, P_{io} , is detected when the signal amplitude passes the threshold. In this way, the cessation of previous burst activity and the onset of the next burst activity based on the current minimum-valued segment are roughly detected. Thus, the interval of P_{ic} and P_{io} will be a coarse definition of the background segment and the interval of P_{io} and $P_{(i+1)c}$ or $P_{(i-1)o}$ and P_{ic} will be a coarse segment of the burst activity. Even though this method can only approximately discriminate the boundary between the burst activity and the background, it provides an explicit, robust determination where the burst activity is. The magnitude of the number, k , ranges in our experiments from 1.5 to 4.0 and is empirically determined according to the variability of the background segment. It was found by running the segmentation system and by visually checking whether or not all burst

activities are correctly detected for the current values of k . Usually the higher value of k is used for P_{ic} detection and the lower value is used for P_{io} detection. The points P_{ic} and P_{io} provide reference points to search for the coarse transitional segment.

2.4.2 Signal Segmentation

The signal segmentation is performed breath by breath. The three segments (burst activity, transitional, and background segment) are classified by local extrema (local minimum and maximum) on the moving time averaged signal, and both zero-crossing points and local extrema on its derivative. The coarse and fine-tuned transitional segments are also determined in this procedure.

Figure 2.22 shows how to determine the transitional segments. The first signal is the original DEMG signal. The second and the third are DEMG moving time averages of window length of 21 and 81 respectively. The fourth channel is a derivative of the third. The “central finite difference” is used for the derivative with equation

$$f'(x_i) = \frac{f(x_i + h) - f(x_i - h)}{2h}$$

Clearly, as the step size, h , gets smaller, the secant line approaches a tangent to the curve at the point x_i . A more accurate estimate of the derivative could be obtained by using a smaller step size or a higher order approximation. However, the lower-order approximation is sufficient because we need to detect only the maximum slope of the MTA signal to derive the transitional segment easily. The step size of $h=5$ points was empirically chosen to smooth the derivative.

For the determination of the *coarse transitional segment*, the third and fourth signals are needed. Zero-crossings, Z_{i1} and Z_{i2} , in the derivative signal are found by searching forward and backward from the thresholding point, P_i (P_{io} in the figure). The minimum-valued point on MTA81 signal, T_{i1} , is the point which is detected by searching from the zero-crossing (Z_{i2}) near the background to another zero-crossing (Z_{i1}) far from the background. Sometimes T_{i1} coincides in time with Z_{i2} . The coarse transitional

segment is defined as the interval between T_{i1} and Z_{i1} . The step to find T_{i1} could be omitted but, in some cases, T_{i1} appreciably decreases the length of the coarse transitional segment.

The transitional segment can be thought of as the estimation of the boundary with an imprecision equal to half the segment length. Therefore, it makes sense to find ways to shorten the segment length. We propose another transitional segment, the *fine transitional segment*, which is smaller than the coarse transitional segment. We need temporal (or local) information within the coarse transitional segment to find the fine transitional segment. The local minimum, M_1 , on MTA21, which is the second signal in the figure, is found in the coarse transitional segment and then the mean value of a duration of 11 samples, $seg1$ in the figure, is calculated which will be a new reference level for determining the fine transitional segment. Another point, N_1 , which is less than the new reference level is found by searching from the local minimum M_1 to Z_{i1} . If it exists, N_1 will be an extremity of the fine transitional segment. Otherwise, M_1 is the limit of the fine transitional segment. The searching continues in order to determine another limit, P_1 , until a peak is reached and the peak amplitude is less than $10.0 \cdot (\text{mean of } seg1)$. If the peak is not found, $P_1 = Z_{i1}$. If both the peak, P_s between Z_{i1} and Z_{i2} in last signal, is less than P_1 , and P_s is greater than M_1 or N_1 in time, then P_1 is set to $P_s + (P_1 - P_s) \cdot (3/4)$. Finally the fine transitional segment is determined as $P_1 - N_1$ or $P_1 - M_1$. Thus, the transitional segments are determined based on the combination of the points, Z_{i1} , Z_{i2} and T_{i1} for the coarse transitional segment, and M_1 , N_1 , P_1 , P_s and Z_{i1} for the fine transitional segment.

The above explanation refers to the determination of the transitional segments for onset. This method is the same for cessation. The length of the transitional segments may vary depending upon the slope of the averaged signal. Sharper signal changes result in a shorter length of the transitional segments and vice versa. Since all boundary estimates will be chosen in the coarse transitional segment it is important to detect the transitional segment without high specificity, and the outline method has the potential to do so.

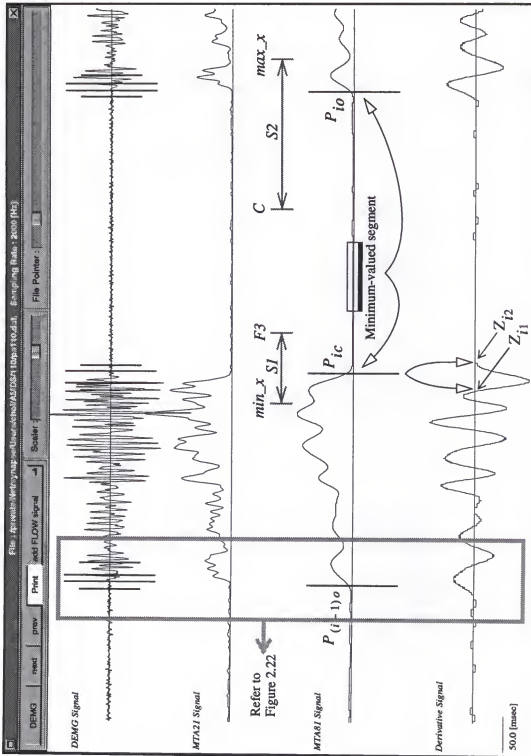


Figure 2.21. Graphical description for recognition of the burst activity

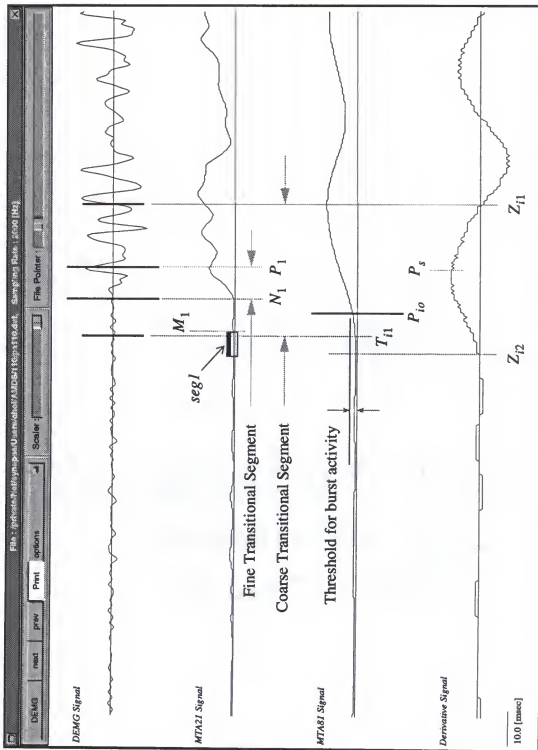


Figure 2.22. Determination of the transitional segments

2.5 Detection of Boundary Candidates

In the previous sections the transitional segment was defined. In this phase of the algorithm, all probable boundary candidates will be extracted from within the coarse transitional segment. The explicit discrimination of the burst activity from the background segment can be done by an energy-based concept. The energy function is computed by adding the square of the signal in a fixed, moving window. The issue is how to enhance the burst activity with respect to the background or tonic activity. The idea to detect the boundary candidates follows an enhancement of the burst activity using the energy function which will improve the timing accuracy of boundary threshold.

One difficulty with the energy function is that it is very sensitive to large signal levels since the signal amplitude enters the computation as a square, thereby emphasizing large sample-to-sample variations in the signal. When the energy function is computed on the raw EMG signal, there are a lot of possible boundary candidates due to signal fluctuations. Many boundary candidates are determined and most of them are useless. A simple way to alleviate sensitivity to signal fluctuations is to take the energy function on the moving time averaged signal instead of the original EMG signal. Thus, a combination of the moving time average of the original signal and the short time energy of the MTA is proposed to realize the energy contour.

2.5.1 Signal Enhancement - Short Time Energy (STE)

It is observed that the amplitude of the burst activity varies appreciably with time. In particular, the amplitude of the background segments is generally much lower than that of the burst activity segments. The short time energy of the EMG signal provides a convenient representation that reflects these amplitude variations. In general, the short-time energy is defined as

$$E_n = \frac{1}{2N+1} \sum_{m=-N}^N (x(m) \cdot w(n-m))^2$$

This expression can be written as

$$E_n = \frac{1}{2N+1} \sum_{m=-N}^N x^2(m) \cdot h(n-m)$$

where $h(n) = w^2(n)$ and the symbol \cdot means convolution.

That is, the signal $x^2(n)$ is filtered by a linear filter with impulse response, $h(n)$, or equivalently the window, as given by the above equation. The choice of $h(n)$ determines the nature of the short time energy representation. In this research, rectangular windows of several length are selected.

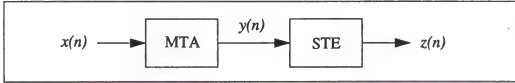
$$h(n) = \begin{cases} 1, & -N \leq n \leq N \\ 0, & \text{otherwise} \end{cases}$$

To see how the choice of the window affects the short time energy, let us observe that if $h(n)$ is very long, and of constant amplitude, E_n will change very slowly with time and thus will not adequately reflect the changing properties of the muscle signal. Such a window would be the equivalent of a very narrowband lowpass filter. Clearly, what is desired is some lowpass filtering but not so much that the output is constant; that is, we want the short-time energy to reflect the amplitude variations of the muscle envelope. It is desirable to have a short duration window (impulse response) to be responsive to rapid amplitude changes, but a window that is too short will not provide sufficient averaging to produce a smooth energy contour. This implies that no single value of N is entirely satisfactory for all patterns of the EMG signal encountered in practice.

The major significance of E_n is that it provides a basis for distinguishing the burst activity segments from the background segments because it can significantly emphasize the slope change of the amplitude during the transition. Usually, the values of E_n for the background segments are quite smaller than those for the burst activity segments. Thus, the energy function can be used to locate the approximate time at which the burst activity starts and ends. For the EMG signal of very high quality (high signal-to-noise ratio), the energy function can be used directly for coarse discrimination.

2.5.2 Combination of MTA and STE

By assuming that the rectangular window is used, mathematical expressions of a combination of the MTA and STE are given by



$$y(m) = \frac{1}{K_1} \sum_{j=m-l}^{m+l} x(j)$$

$$\begin{aligned} z(n) &= \frac{1}{K_2} \sum_{k=n-i}^{n+i} y^2(k) \\ &= \frac{1}{K_1 K_2} \sum_{k=n-i}^{n+i} \left[\sum_{j=k-l}^{k+l} x(j) \right]^2 \\ &= \frac{1}{K} \sum_{k=n-i}^{n+i} \left(\sum_{j=k-l}^{k+l} x(j) \right) \left(\sum_{j=k-l}^{k+l} x(j) \right) \end{aligned}$$

The operation of the MTA and STE can be stated as the weighted summation of both the square of each point and multiplication of adjacent points with different window length. For example, consider the window length $N=3$ for both the MTA and the STE, $Kz(n)$ is

$$\begin{aligned} Kz(n) &= x^2(-2) + 2x^2(-1) + 3x^2(0) + 2x^2(1) + x^2(2) + \\ &\quad 2[x(-2)x(-1) + 2x(-1)x(0) + 2x(0)x(1) + x(1)x(2) + \\ &\quad x(-2)x(0) + x(-1)x(-1) + x(0)x(2)] \end{aligned}$$

The only variable in this expression is the window length for both the MTA and the STE. One cascade of MTA and STE blocks with fixed window length gives rise to one boundary point. Due to the fuzziness of the boundary, we propose to determine several boundary points in the transitional segment. Thus, a parallel structure of the cascade of MTA/STE with different windows is proposed to detect several boundary points because

different window lengths generate different signal slopes. These boundary points were called "boundary candidates." The block diagram of the parallel structure is shown in Figure 2.23.

The question is how we can determine several window lengths for both the MTA and the STE. Generally speaking, the bandwidth is inversely proportional to the length of the window. For a small window length of both algorithms, the output of the cascade fluctuates rapidly so it detects amplitude variations of the signal slope that are too small. For a large window length, the response will change too slowly, so it cannot follow the signal fluctuation. There is no criteria to fix the window length of both the MTA and the STE to detect probable boundaries for all types of transitional signal patterns. Instead, based on the following general rules for choosing the window length, we can specify several representative combinations of window length.

- (1) For slow slope change, a large window length of the MTA is appropriate (assuming a fixed window length of the STE).
- (2) For sharp slope change, a small window length of the MTA is appropriate (assuming fixed window length of the STE).

These rules are the same for the fixed window length of the MTA and variable window length of the STE.

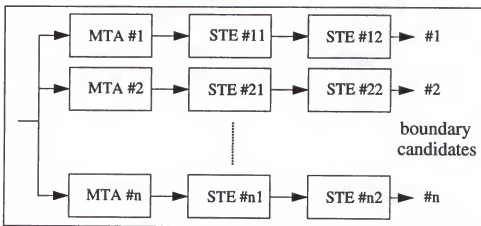


Figure 2.23. Parallel structure of the serial combination

However, experimentation has demonstrated that the slope change is more sensitive to the MTA window than that of the STE because not only is the STE computed after the MTA signal, but the STE is also much more sensitive to the signal variation. With these aspects for the determination of the window length in mind, we have found that a suitable window length, N is on the order of 1 to 71 for the MTA and 1 to 11 for the STE in order to cover all patterns of transitional signal changes. In this research, five parallel structures of the cascade are empirically determined with the following window lengths: 1-1-1, 2-2-2, 31-7-9, 51-9-11, and 71-2-2. The first two combinations are used to detect relatively small variations of signal slope changes while the rest are for large signal changes.

In section 3.4, the window length of 81 was used for the burst activity identification because we need global information of the signal. For detection of the boundary candidates, a smaller window length is useful since they are determined by detecting local signal changes in the transitional segment. However, it is important to get estimates that are uniformly distributed around the true boundary. So we should always choose long and short windows. If the estimates are biased, then there is no way the neural network can provide an unbiased estimate.

The above algorithm possesses somewhat a self-adaptation of level thresholding to determine fiducial boundaries. The ensemble of the detected boundaries from the structure of the cascade represents all probable boundary candidates for almost all the cases. Increasing the number of these fiducial candidates only reduces the probability of boundary detection error. These boundary candidates are sorted through pattern classification which will be discussed in next chapter.

CHAPTER 3

BOUNDARY ESTIMATION USING ARTIFICIAL NEURAL NETWORK

With the technique presented in the previous chapter, a large number of boundary candidates in the transitional segment are determined. In this chapter, two problems are tackled. First, all boundary candidates are rated; that is, the boundary probability for each candidate will be computed in a consistent way. An artificial neural network as a statistical classifier is proposed for this goal. The chapter also describes how the neural network functions to estimate the boundary candidates. Second, based on the most probable boundary, the final boundary point will be found for a given transitional segment.

3.1 Introduction

Except for sharp transitional patterns (low amplitude background and large burst activity), it is not an easy task to delineate a separation between the background and the burst activity both in terms of the signal analysis and the physiological viewpoints. Unfortunately, ill-defined transitions (called here fuzzy) are very common in respiratory EMG patterns. For fuzzy transitional signals, different experts may select several boundaries with different degrees of certainty. Even the same expert may select different boundary points when presented with the same data on different occasions. This fact may be interpreted as saying that in a given transitional segment, there are several possible boundaries that the experts prefer due to their experience and school. For the automated system, assuming that the boundary candidates have been detected using the energy functions as explained in chapter 2, the issue is to compute a discriminant function to estimate consistently the boundary candidates based on the same criterion.

The useful information that can be obtained from the signal to help us rate the

boundaries is: (1) features extracted from the boundary point and (2) physiological knowledge from the multichannel-supported data. In a broad sense, decision making can be divided into two approaches: symbolic and numeric processing. A rule-based approach could be the representative method in the symbolic processing while pattern recognition (specifically, classical methods such as minimax, k-means, and maximum likelihood, and artificial neural network) fits the numeric processing.

The rule-based system requires extensive rules for decision making according to signal features and inferences from the physiological knowledge. Since the boundary decision is not well structured domain, it is not simple to make the rules explicit without knowing every condition of the signal shape. If the matching rules do not exist for the particular boundary under analysis, the system might fail. Another problem in the rule-based approach is the inherent weakness of physiological inferences from the raw signal since the transitional segment is too variable. In other words, the physiological interpretation gives only general information to help the decision of the precise boundary. In the pattern classification approach, a promising methodology is based on artificial neural networks (ANN). Although much of the interest in neural nets has arisen in their similarities with biological information processing, their popularity has been enhanced by the fact that, in principle, they are relatively easy to use. They have been found useful in the area of SONAR target [Go88], speech recognition [Lib87], EEG sleep staging [Pri89], metal surface classification [Br90], robotics [Ed87], etc.

The ANN learns the classification problem through the presentation of sufficient and meaningful training data sets (or examples). The system parameters determined by the network training are able to generalize (or infer) characteristics of new data not belonging to the training set. The system training through examples means that the network does not require any mathematical or statistical knowledge to derive the discriminant function. Considering that most boundaries have dynamic conditions with vague definition, the ANN seems a better way to handle indistinct boundaries rather than the rule-based system in

which clear rules for the boundary inferences are hard to explicate, or even parametric pattern recognition where the a priori probability of the boundary distributions would be required. Another important feature of the ANN approach is that it can be made to estimate posterior probabilities directly.

The ability of the neural net to have its parameters estimated from examples has been considered one of its major strengths. Such an ability is of course very important, but because of the highly nonlinear nature of neural nets, the large amount of computation required for training (or learning) and the sensitivity of the training methods to local minima must be considered the weakest point of neural nets.

For highly variable signals, the boundary decision cannot be expected to be linear; i.e. the discriminant function will very likely be a high order polynomial. Nonlinear discriminant functions which map input vectors to the decision space can be obtained with a multilayered ANN. Thus, the multilayered ANN which is a feed-forward network with one or more layers between the input and output nodes will be used here as the pattern classifier. The number of layers and the number of nodes in the hidden layers are determined based on experimental results.

The success of the ANN implementation relies on good training. For the ANN to infer precise boundaries for variable signal patterns, it should be trained by diverse examples, spanning the quite slowly changing boundaries up to sharply changing ones. Thus, training sets which contain representative boundaries must be collected from the original EMG signal. The human expert will visually choose the training set. During adaptation, the desired output values are set to 1 for boundary class and 0 for non-boundary class. The ANN output constitutes the boundary probabilities, indicating how close the input vectors are to the true boundary.

The input to the net is directly extracted from the raw EMG signal, not from an averaged signal. If we use the input vector extracted from the average signal, we may lose the time resolution of boundary. The issue is how to estimate the signal property at the

boundary. Since a major feature between the burst activity and the background is amplitude, we propose a modified amplitude information ("*duration-slope*" of a half-wave defined in section 3.3) as the input vector. To reduce the dynamic range of the transitional signal change, the input vector is reconstructed from the original signal before classification. That is, duration-slopes of half waves around the boundary candidates are normalized and quantized. The boundary probabilities of all boundary candidates in the transitional segment are computed with the ANN. The block diagram of boundary estimation is shown in Figure 3.1. The final boundary is determined within the half-wave of the highest probability, as a point intersecting the background level. Thus, there is only one final boundary in the given transitional segment.

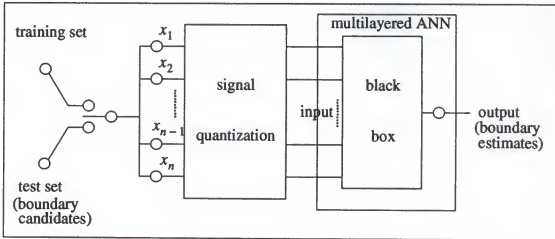


Figure 3.1. Block diagram of boundary estimation

3.2 Neural Network as Pattern Classifiers

It has recently been shown that neural nets, when trained with a desired output of 1 for belonging to a class and 0 otherwise, produce as their output an estimate of the posterior probability of the class. Even though their performance as *a posteriori* probability classifiers varies slightly based on the error criteria (such as least square error and Kullback-Leibler estimator), it has been shown that the neural nets produce *a posteriori*

probability estimates [El90, Ma91, Gi90, and Wh89].

Addressed here are properties of the probability estimation of the neural nets, especially the ability of the neural nets to estimate *a posteriori* probabilities.

3.2.1 Posterior Probability Estimation

3.2.1.1 Classification

Let us assume that the input vector, \mathbf{x} , belongs to one of M classes, c_i , $1 \leq i \leq M$. The main objective in pattern recognition is to decide which of the M classes the vector \mathbf{x} belongs to. Typically, we use a decision rule that takes \mathbf{x} as input and gives the class as output. The decision rule can be based on some form of discriminant functions that is a function of \mathbf{x} . A more general decision rule that guarantees the minimum classification error is what is known as the Bayes decision rule, namely, choosing the class c_i for which the *a posteriori* probability, $P(c_i|\mathbf{x})$, is maximum for a given \mathbf{x} . All one requires, therefore, is the ability to estimate $P(c_i|\mathbf{x})$ for each of the classes.

Instead of estimating $P(c_i|\mathbf{x})$ directly, most methods rely on Bayes' theorem

$$P(c_i|\mathbf{x}) = \frac{p(\mathbf{x}|c_i) P(c_i)}{p(\mathbf{x})}$$

where $p(\mathbf{x}|c_i)$ is the conditional density (also known as the likelihood) and $P(c_i)$ is the *a priori* probability of the class c_i . Now, since $p(\mathbf{x})$ is the same for all classes, a decision rule based on maximizing the joint density, $p(\mathbf{x}, c_i) = p(\mathbf{x}|c_i) P(c_i)$, is equivalent to maximizing $P(c_i|\mathbf{x})$. The computation of $P(c_i)$ is simple and the literature abounds with methods of estimating likelihoods of $p(\mathbf{x}|c_i)$ [To74 and Du73]. The neural nets compute $P(c_i|\mathbf{x})$ directly as explained in next section.

3.2.1.2 Least squares error measure for posterior probability estimation

This section demonstrates that the least squares error measure for neural networks is equivalent to the expected value of the square of the difference between the output of the network and the posterior probability of a class given the input.

The conditional probability law plays an important role in interpreting neural nets in terms of statistical viewpoint. Let's define the conditional expectation of the target output, t , for the given input, x , denoted $E(t|x)$. This conditional expectation gives the value of t that will be realized on average given a particular realization for x , i.e., $g(x) = E(t|x)$. The actual realization of t will almost always differ from $g(x)$. Thus, "expectational error" is defined as $\varepsilon = t - E(t|x)$. So we can rewrite $t = E(t|x) + \varepsilon = g(x) + \varepsilon$. By definition of ε and by the properties of conditional expectation, it follows that $E(\varepsilon|x) = 0$. That is, the average expectational error given any realization of x is zero.

For this part of the discussion we assume that we have a 2-class problem (class c and \bar{c}) with a single output neural network, $y(x, \theta)$, where the dependence of y on the parameters, θ , of the network is made explicit. A popular method of training the neural net is to determine the parameters that minimize the mean (or expected) squared error between the network output, y , and a target function, $t(x)$. The least square measure between the target function and the network output is given by

$$E_{LS} = \frac{1}{N} \sum_{i=1}^N [y(x, \theta) - t(x)]^2$$

$$t(u) = \begin{cases} a, u \in c \\ b, u \in \bar{c} \end{cases}$$

where N is the number of training vectors. The $t(x)$ is the desired output which is equal to 1 if the input belongs to class c , and equal to 0 if the input does not belong to class c . If the number of training is large, the summation of the above equation converges to the expected value.

$$\begin{aligned} E_{LS} &= E[y(x, \theta) - t(x)]^2 \\ &= E[y(x, \theta) - g(x) + g(x) - t(x)]^2 \\ &= E[y(x, \theta) - g(x)]^2 + 2E([y(x, \theta) - g(x)][g(x) - t(x)]) \\ &\quad + (E[g(x) - t(x)]^2) \end{aligned}$$

Based on definition of $\varepsilon = t - E(t|x)$, $g(x) = E(t|x)$, and conditional expectation

theory of $E[m_1(x)m_2(y)] = E[E\{m_1(x)m_2(y)|x\}] = E[m_1(x)E\{m_2(y)|x\}]$,

the second part of the last equality becomes

$$\begin{aligned} E([y(x, \theta) - g(x)][g(x) - t(x)]) &= E([y(x, \theta) - g(x)]\varepsilon) \\ &= E(E([y(x, \theta) - g(x)]\varepsilon|x)) = E([y(x, \theta) - g(x)]E[\varepsilon|x]) = 0 \end{aligned}$$

by the law of expectations and properties of ε . The least square error measure can be rewritten as:

$$\begin{aligned} E_{LS} &= E[y(x, \theta) - g(x)]^2 + E[g(x) - t(x)]^2 \\ &= E[y(x, \theta) - E(t|x)]^2 + E[E(t|x) - t(x)]^2 \end{aligned}$$

Since in the above equation, the second term does not depend on the parameters θ , minimizing the equation is equivalent to minimizing

$$E_{LS} = E[y(x, \theta) - E(t|x)]^2$$

But,

$$E(t|x) = aP(c|x) + bP(\bar{c}|x) = (a-b)P(c|x) + b \quad (\text{Eq. 1})$$

since $P(\bar{c}|x) = 1 - P(c|x)$. For $a=1$ and $b=0$, $E(t|x) = P(c|x)$ and the above equation reduces to

$$E_{LS} = E[y(x, \theta) - P(c|x)]^2$$

Therefore, by minimizing the E_{LS} between the output, y , and the target function, t , with $a=1$ and $b=0$, the neural net gives as output a least squares estimate of the posterior probability, $P(c|x)$. This result gives us an explicit expression for the manner in which the neural net provides an estimate of the posterior probability. This result has been derived in various way by different authors [Wh89, El90, Gi90 and Ma91].

A number of researchers have advocated using values of a and b different from 1 and 0. Eq 1 shows that setting a and b other than 1 and 0 results in a biased $P(c|x)$. However, if the neural net is being used only for classification, then a bias of this kind will not affect the results. But, there are applications for which the probability values are important, in which case $a=1$ and $b=0$ should be used. The reason that researchers have used other values of a and b is because of training difficulties when the classes are

perfectly separable, i.e., in cases where $P(c|\mathbf{x})$ is identically 0 or 1 in different regions. Clearly, for such problems, a nonlinearity that goes to 0 and 1 for finite values is needed. For general problems where the classes are not separable, values of $a=1$ and $b=0$ do not cause any training difficulties.

3.2.1.3 Conditions as probability estimation

It has been shown that the neural net trained with the least square error criterion provides an approximation to the posterior class probabilities, $P(c|\mathbf{x})$. The amount of an approximation to the probability estimation can be achieved depending on the training data and also on the structure of the network. It means that the outputs must obey the rules of probability, i.e.,

$$(1) 0 \leq y_i(n) \leq 1, \text{ for all } c_i \text{ and } \mathbf{x}(n)$$

$$(2) \sum_{i=1}^M y_i(n) = 1, \text{ for each } \mathbf{x}(n)$$

The first condition can be met easily by having the output node of nonlinearity, i.e., a sigmoid type function

$$y_k = f(\text{net}_k) = \frac{1}{1 + e^{-\text{net}_k}}$$

where net_k is the input to the nonlinearity (see Figure 3.2(a)).

The second condition states that the sum of the outputs for each input vector must add up to unity if the number of outputs equals the number of classes, or the sum must be less than unity if the number of outputs is one less than the number of classes. Otherwise, the outputs are not probabilities, even if they obey the first condition. Minimizing E_{LS} does not guarantee that the second condition will be met. A constraint on the structure of the network must be placed to ensure the second condition.

To test that the ANN output is in fact a probability density function, two different sets of 50 inputs are tested using the topology ($M=2$) shown in Figure 3.2(b). The result is shown in Table 3.1. One test set labelled "case1" in the table was trained so well that the

system error was small whereas in the other case, "case 2" was not well trained. Nevertheless, the sum of output was almost equal to one. For case 1, the average mean of the sum of outputs is 1.000004, its standard deviation 0.0001909, its maximum 1.000683, and its minimum 0.999103. For case 2, associated values are 1.0021351, 0.0046001, 1.020825, and 0.999866. The table also shows that all outputs are in the range of [0,1].

3.2.1.4 Neural net as a discriminant

Figure 3.2(a) shows a representation of the neural net where the mapping from the input, x , to h contains the effects of all hidden layers, and the output layer with a single output is shown. The hidden layers can be thought of as implementing a set of feature extractors, $h_i(x)$, $1 \leq i \leq p$, mapping the n -dimensional input space x into a p -dimensional space h . The last stage then implements a hyperplane ($h \cdot w$) which separates class c from its complement \bar{c} . In other words, the hidden layers map the boundary between c and \bar{c} in x space to a linear boundary in h space. Therefore, we can view the neural nets as implementing a mapping from one feature space into another where a class can be separated from the rest of classes by a linear discriminant.

Note that this interpretation is independent of what $f(net_k)$ is, as long as it has a monotonic relation to net_k . One method that enhances the ability of the neural net to perform the required boundary mapping to a plane may transform the input into a much higher dimensional space where the classes are separable by planes. It is noticeable that while the linear discriminant is the best discriminant for two Gaussians with equal variance, a single-node network with a sigmoid nonlinearity gives the exact posterior probability as its output.

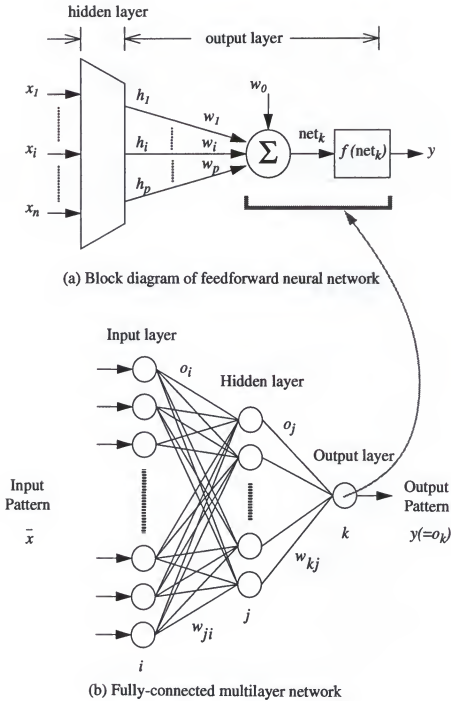


Figure 3.2. The structure of the neural network

Table 3.1. Test of conditions for probability estimation

	output, y1	output, y2	y1+y2	output, y1	output, y2	y1+y2	output, y1	output, y2	y1+y2
B1	0.998009	0.001814	1.000000	0.999983	0.000017	1.000000	0.999983	0.000017	1.000000
B2	0.997157	0.002821	0.999978	0.999978	0.000022	1.000000	0.999978	0.000022	1.000000
B3	0.996435	0.003565	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B4	0.995719	0.004281	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B5	0.995000	0.005000	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B6	0.994281	0.005719	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B7	0.993565	0.006435	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B8	0.992850	0.007157	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B9	0.992135	0.007871	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B10	0.991419	0.008586	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B11	0.990704	0.009301	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B12	0.989989	0.010015	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B13	0.989274	0.010729	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B14	0.988559	0.011443	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B15	0.987844	0.012157	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B16	0.987129	0.012871	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B17	0.986414	0.013586	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B18	0.985699	0.014300	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B19	0.984984	0.015015	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B20	0.984269	0.015729	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B21	0.983554	0.016443	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B22	0.982839	0.017157	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B23	0.982124	0.017871	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B24	0.981409	0.018586	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B25	0.980694	0.019300	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B26	0.979979	0.020015	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B27	0.979264	0.020729	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B28	0.978549	0.021443	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B29	0.977834	0.022157	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B30	0.977119	0.022871	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B31	0.976404	0.023586	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B32	0.975689	0.024300	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B33	0.974974	0.025015	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B34	0.974259	0.025729	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B35	0.973544	0.026443	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B36	0.972829	0.027157	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B37	0.972114	0.027871	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B38	0.971399	0.028586	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B39	0.970684	0.029300	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000
B40	0.969969	0.030015	0.999999	0.999999	0.000001	1.000000	0.999999	0.000001	1.000000

Case 1

Case 2

3.2.2 Network Learning Algorithm

Backpropagation learning algorithm for the feedforward net is used to train the network. The semilinear feedforward net as reported by Rumelhart [Ru86] has been found to be an effective system for learning the discriminants of patterns from a body of examples. It is similar to the architecture of the α -Perceptron proposed by Rosenblatt [Ro62] except that there are hidden layers. Figure 3.2(b) shows the fully-connected multilayer network which is also used in this study.

The outputs of nodes, o_i or o_j in Figure 3.2, in one layer are transmitted to nodes in another layer through links, w_{ji} or w_{kj} , that amplify or attenuate such outputs through weighting factors. Except for the input layer nodes, the net input, net_k , to each node is the sum of the weighted outputs of the nodes in the prior layer. Each node is activated in accordance with the input to the node which is the output of the prior layer, the activation function of the node, and the bias of the node.

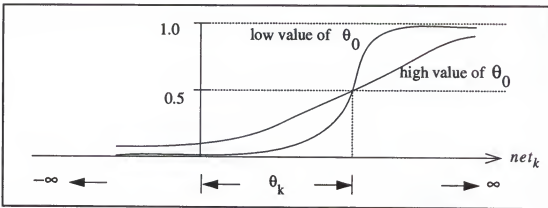


Figure 3.3. Sigmoidal activation function

For a sigmoidal activation function used in this research,

$$f(net_k) = \frac{1}{1 + e^{-(net_k + \theta_k)/\theta_0}}$$

where θ_k serves as a threshold or bias. The effect of a positive θ_k is to shift the activation function to the left along the horizontal axis, and the effect of θ_0 is to modify the shape of

the sigmoid. A low value of θ_0 tends to make the sigmoid function take on the characteristics of a threshold-logic unit (TLU), whereas a high value of θ_0 results in a more gently varying function. These effects are illustrated in Figure 3.3.

The components of an input pattern, \bar{x}_p in pattern p , constitute the inputs to the nodes in layer i . The net inputs to nodes in layers j and k are

$$net_j = \sum w_{ji} o_i \text{ and } net_k = \sum w_{kj} o_j$$

The outputs of nodes j and k are

$$o_j = f(net_j) = \frac{1}{1 + e^{-(net_j + \theta_j)/\theta_0}} \text{ and } o_k = f(net_k)$$

In the learning phase of training the network, the input pattern, \bar{x}_p , is presented and then the net adjusts a set of weights in all the connecting links and also all the thresholds in the nodes so that the desired outputs, t_{pk} , are obtained at the output nodes. Once this adjustment has been accomplished by the net, another pair of \bar{x}_p and t_{pk} is presented and the net learns that association also. Finally the net finds a single set of weights and biases that will satisfy all the (input, output) pairs presented to it. This process can pose a very strenuous learning task and is not always easily accomplished.

Generally, for the pattern p , the outputs, o_{pk} , will not be the same as the target or desired values, t_{pk} . For each pattern, the square of the error in output is

$$E_p = \frac{1}{2} \sum_k (t_{pk} - o_{pk})^2$$

where the factor of one-half is inserted for mathematical convenience. In the generalized delta rule for learning the weights and biases, the procedure for learning the correct set of weights is to adjust the weights so that the error, E_p , is reduced as rapidly as possible. The corrections to the weights are made sequentially, on the basis of the learning to be carried out for the sequence of the patterns, one at a time.

Omitting the p subscript, the error measure can be rewritten as

$$E = \frac{1}{2} \sum_k (t_k - o_k)^2$$

For parameters in the network to converge toward improved values for the weights

and thresholds, incremental changes Δw_{kj} proportional to $(-\partial E) / (\partial w_{kj})$ are made,

$$\Delta w_{kj} = -\eta \cdot \frac{\partial E}{\partial w_{kj}}$$

However, the error, E , is expressed in terms of the outputs, o_k . The partial derivative, $(\partial E) / (\partial w_{kj})$, can be evaluated using the chain rule

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial net_k} \frac{\partial net_k}{\partial w_{kj}} = \frac{\partial E}{\partial net_k} \left(\frac{\partial}{\partial w_{kj}} \sum w_{kj} o_j \right) = \frac{\partial E}{\partial net_k} o_j$$

If we define

$$\delta_k = -\frac{\partial E}{\partial net_k}$$

then

$$\Delta w_{kj} = \eta \delta_k o_j$$

To compute $\delta_k = -(\partial E) / (\partial net_k)$, the partial derivative using the chain rule in terms of two factors, one expressing the rate of change of error with respect to the output, o_k , and the other expressing the rate of change of the output of the node k with respect to the input to that same node,

$$\delta_k = -\frac{\partial E}{\partial net_k} = -\frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial net_k}$$

The two factors are obtained as follows:

$$\frac{\partial E}{\partial o_k} = -(t_k - o_k) \text{ and } \frac{\partial o_k}{\partial net_k} = \frac{\partial}{\partial net_k} (f_k(net_k)) = f'_k(net_k)$$

Therefore,

$$\delta_k = (t_k - o_k) f'_k(net_k)$$

for any output-layer node k ,

$$\Delta w_{kj} = \eta (t_k - o_k) f'_k(net_k) o_j = \eta \delta_k o_j$$

The weights of hidden layers do not affect output nodes directly. But it can be written in the same way

$$\begin{aligned} \Delta w_{ji} &= (-\eta) \frac{\partial E}{\partial w_{ji}} = (-\eta) \frac{\partial E}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}} = (-\eta) \frac{\partial E}{\partial net_j} o_i \\ &= \eta \left(-\frac{\partial E}{\partial o_j} \cdot \frac{\partial o_j}{\partial net_j} \right) o_i = \eta \left(-\frac{\partial E}{\partial o_j} \right) f'_j(net_j) o_i = \eta \delta_j o_i \end{aligned}$$

However, the factor $(\partial E) / (\partial o_j)$ cannot be evaluated directly. Instead, it can be written

in terms of quantities that are known and other quantities that can be evaluated.

$$\begin{aligned} -\frac{\partial E}{\partial o_j} &= -\sum_k \frac{\partial E}{\partial net_k} \frac{\partial net_k}{\partial o_j} = \sum_k \left(-\frac{\partial E}{\partial net_k} \right) \frac{\partial}{\partial o_j} \sum_m w_{km} o_m \\ &= \sum_k \left(-\frac{\partial E}{\partial net_k} \right) w_{kj} = \sum_k \delta_k w_{kj} \end{aligned}$$

Therefore,

$$\delta_j = f'_j(net_j) \sum_k \delta_k w_{kj}$$

That is, the deltas at an internal node can be evaluated in terms of the deltas at the next layer.

Thus, starting at the output layer, δ_k is first evaluated, and then the error computation is propagated backward to the input layers. Summarizing, using the additional subscript p to denote the pattern number, the incremental weight changes of nodes in hidden layers and the output layer can be expressed as follows:

$$\Delta_p w_{ji} = \eta \delta_{pj} o_{pi}, \text{ where } \delta_{pj} = f'_j(net_{pj}) \sum_k \delta_{pk} w_{kj} \text{ for hidden layers}$$

$$\Delta_p w_{kj} = \eta \delta_{pk} o_{pj}, \text{ where } \delta_{pk} = (t_{pk} - o_{pk}) (f'_k(net_{pk})) \text{ for the output layer}$$

In particular, if

$$o_j = \frac{1}{1 + \exp(-(\sum_i w_{ji} o_i + \theta_j))}$$

then

$$f'_j(net_j) = \frac{\partial o_j}{\partial net_j} = o_j(1 - o_j)$$

Note that the thresholds, θ_j , are learned in the same manner as are the other weights.

We simply imagine that θ_j is the weight from a unit that always has an output value of unity. Also note that the derivative, $(\partial o_j) / (\partial net_j)$, equal to $o_j(1 - o_j)$, reaches its maximum for $o_j=0.5$ and, since $0 \leq o_j \leq 1$, it approaches its minima as o_j approaches zero or one. Since the change in weight is proportional to this quantity, it is clear that weights that are connected to units in their midrange are changed the most. The weights change rapidly under those conditions, and this feature probably contributes to the stability of the learning procedure.

It is important to note that, for the activation function, a node cannot have output values of 1 or 0 without infinitely large positive or negative weights. Therefore, in the

learning mode, the values of 0.9 and 0.1 might suffice for specifying binary target output values.

In learning w_{ji} , it is good practice to calculate $\Delta_p w_{ji}$ for each pattern in the training set, and to take

$$\Delta w_{ji} = \sum_p \Delta_p w_{ji}$$

The learning procedure therefore consists of the net starting off with a random set of weight values, choosing one of the training-set patterns, and, using that pattern as input, evaluating the outputs in a feedforward manner. The errors at the output(s) generally will be quite large, which necessitates changes in the weights. Using the backpropagation procedure, the net calculates $\Delta_p w_{ji}$ for all the w_{ji} in the net for the particular pattern, p . This procedure is repeated for all the patterns in the training set to yield the resulting Δw_{ji} for all the weights for that one presentation. The corrections to the weights are made and the outputs are again evaluated in feedforward manner. Discrepancies between actual and target output values again result in the evaluation of weight changes. After a complete presentation of all patterns in the training set, a new set of weights is obtained and new outputs are again evaluated in a feedforward manner.

The net can track the system error as well as the errors for individual patterns. In a successful learning exercise, the system error will decrease with the number of iterations, and the procedure will converge to a stable set of weights, which will exhibit only small fluctuations in value as further learning is applied.

There are several other issues to keep in mind when such nets are implemented. There is, for example, the question of how the value of η is to be chosen. This is a common problem in all steepest-descent methods for locating minima of functions. As might be expected, a large η corresponds to rapid learning but might also result in oscillations. One way to reduce the learning rate without leading to oscillation is to modify the back propagation learning rule to include the *momentum* term. That is, instead of $\Delta w_{ji} = \eta \delta_j o_i$,

$$\Delta w_{ji}(n+1) = \eta \delta_j o_i + \alpha \Delta w_{ji}(n)$$

where the quantity $(n+1)$ is used to indicate the $(n+1)$ th step, and α is a proportionality constant. The second term in the above expression is used to specify that the change in w_{ji} at the $(n+1)$ th step should be somewhat similar to the change undertaken at the n th step. In this way, some inertia is built in, and momentum in the rate of change is conserved to some degree. An examination of the system error, E , over a large number of steps in the iterative approach to a solution will generally show that a finite α tends to dampen the oscillations but can also serve to slow the rate of learning. In the automated segmentation system, $\eta=0.7$ and $\alpha=0.9$ are used.

Note that the net must not be allowed to start off with a set of equal weights. If all weights start out with equal values and if the solution requires that unequal weights be developed, the system can never learn [Ru86]. This is because the error is propagated back through the weights in proportion to the values of the weights. This problem is solved in the automated segmentation system by setting random values of the initial weights.

Another issue concerns the question of whether the system might get trapped in some local minimum or even at some stationary point, or perhaps oscillate between such points. Under such circumstances, the system error remains large regardless of how many iterations are carried out. This situation is depicted schematically in Figure 3.4, which shows the system error as a function in weight space. It is desirable for the net to learn the weights w_{minimum} , but the net might get trapped at point w_{local} or at point $w_{\text{stationary}}$.

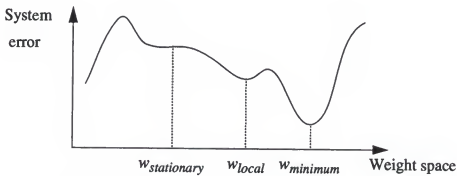


Figure 3.4. Illustration of several possible minima in learning procedure

3.3 Preprocessing for the Network Input

Initially a fixed window of the raw EMG signal around the boundary candidate was used as the input vector. However, the network error did not always decrease, and sometimes it even oscillated for some training sets. The unstable adaptation may be caused by huge variations of input patterns in terms of amplitude changes. For stable and fast adaptation of the network, the raw signal was discretized in fewer levels.

Figure 3.5(a) shows a block diagram for the signal quantization utilized. First, peak points in the fixed window around the boundary candidate are detected. After rejecting noisy peaks, duration-slopes of all line segments are computed and then these are compared with the background level. Duration-slopes which are bigger than the background level are quantized into several discrete levels which are used as input patterns to the neural network.

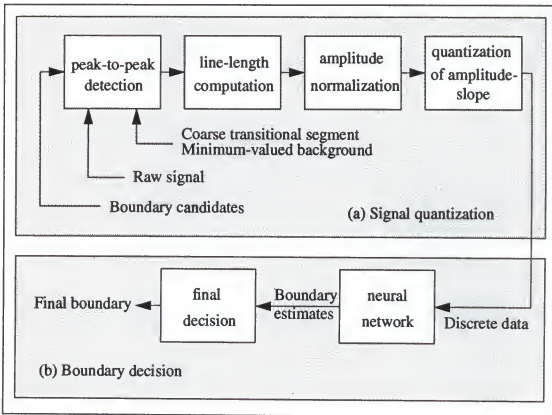


Figure 3.5. Block diagram of signal quantization and boundary decision

As shown in Figure 3.5(b), the ANN attributes *a posteriori* probabilities to the half waves, and the final boundary (a point) is obtained from the half wave that displays the highest *a posteriori* probability. The final boundary is determined based on experimental rules.

3.3.1 Peak-to-Peak Detection

The goal of peak detection is to detect all peaks and recognize the noisy peaks. Thus, the subset of the real peaks, which can be thought to come from the strong MUAP potentials, are found by rejecting the noisy peaks. The noisy peaks can be considered as pairs of adjacent peaks in the background that satisfy certain criteria. Of course, the criteria are dependent on the signals to be processed.

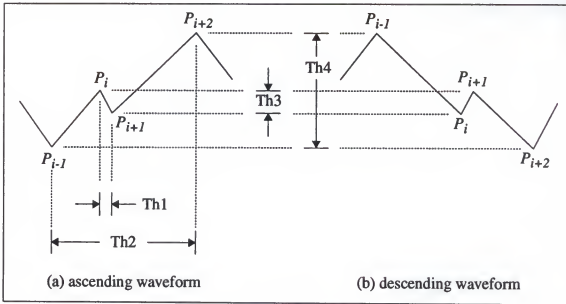


Figure 3.6. Graphical description of noise peaks pair

The description of the noise peak pair is shown in Figure 3.6 [Ch88]. The concept of noisy peaks is as follows: a peak is symbolized by $P(x_p, y_p)$ or simply P_i , where P_i is the i th peak and $i = 1, 2, \dots$. In a flat line the peak can be arbitrarily chosen to be the rightmost point of the line. The peaks are considered as having a left boundary, the extremum of the

previous peak, and a right boundary, the extremum of next peak. Thus, the peak, P_i is always a triplet of points so that $P_i = \{P(x_{i-1}, y_{i-1}), P(x_i, y_i), P(x_{i+1}, y_{i+1})\}$. The signal can be thought to be the sequential connection of these peaks whereas the line segment can be described by a connection of two adjacent peaks.

Based on the observation that most noisy peaks in biological waveforms appear as a pair of adjacent peaks, the peak pairs which satisfy the following criteria are considered as noisy peaks. The criteria for ascending waveform are given with reference to the peak pair (P_{i-1}, P_i) .

$$\text{Criterion 1: } \{|P(x_{i-1}, y_{i-1})| < |P(x_{i+1}, y_{i+1})|\} \text{ and } \\ \{|P(x_i, y_i)| < |P(x_{i+2}, y_{i+2})|\}$$

$$\text{Criterion 2: } Th4 \geq \eta_1 \cdot Th3$$

Criterion 3: if Criterion 2 is not satisfied and

$$\{(\eta_2 \cdot Th3 \leq Th4) \text{ and } (Th4 < \eta_1 \cdot Th3)\}$$

$$\text{Criterion 3.1: } Th1 < \eta_3$$

$$\text{Criterion 3.2: } \{Th1 < \eta_4 \cdot Th2\}$$

The first criterion means that the noisy peak pair should be nested within other larger peaks. The second criterion, amplitude constraint, requires that the magnitude difference between the noisy peaks should be within a certain value. The last criterion, time constraint, requires that the duration between noisy peaks should also be within a certain value. Peaks having an amplitude and duration greater than thresholds which are set by the user cannot be considered as noise; i.e., only peaks which satisfy the above criteria are noisy peaks. The criteria used are empirical and there is no theory from which they can be deduced. The η values used in the automated segmentation system are determined as $\eta_1=10$, $\eta_2=8$, $\eta_3=2$, and $\eta_4=4$ based on observation of the signal. The above criteria can also be directly applied to descending waveforms.

Whenever a new peak is found and the number of peaks is greater than three, the

previous pair of peaks is tested. If the previous peak pair meets the criteria, the algorithm recognizes it as a noisy peak pair, and rejects them. Subsequently the previous peak pair is checked and so on. The nested noisy peak pairs are, therefore, checked and rejected this way until the algorithm does not detect noisy peaks. However, this algorithm cannot reject the noisy peaks when they coincide or are superimposed on real peaks.

3.3.2 Line-length Computation (Duration-Slope)

Even though noise has been reduced using the bandpass filter, the noise in passband or residual noise may exist in the filter output. There are two reasons to compute line-length: first, the final boundary detection is detected in a line segment, so line segments (half waves) are the appropriate features to input to the ANN. Second, ideally a boundary from the background to the burst activity should start at a half wave that is sufficiently steep and clearly distinguished from the background by its amplitude. This is the reason we propose to use as a feature the duration-slope. The computation of the duration-slope based on Figure 3.7 is as follows:

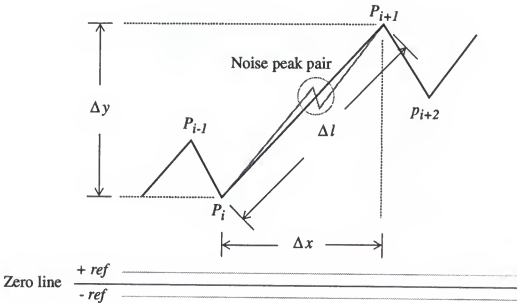


Figure 3.7. Computation of duration-slope

Time-interval computation (Δx):

If the noisy peak pairs are rejected in current line segment

(1) When both P_i and P_{i+1} are greater than zero or both are less than zero

or

(2) When P_i is positive value and P_{i+1} is negative value or vice versa and

$$(P_{y,i+1} - P_{y,i}) > \mu \cdot (P_{x,i+1} - P_{x,i}), \text{ where } \mu=3.$$

$$\text{Time interval at current time } i, \Delta x = (P_{x,i+1} - P_{x,i}) / (\text{noiseno} \cdot 2 + 1)$$

Otherwise

$$\text{Time interval at current time } i, \Delta x = P_{x,i+1} - P_{x,i}$$

Amplitude computation (Δy):

If $\{ (P_{y,i} > \text{ref}) \text{ and } (P_{y,i+1} > \text{ref}) \}$ or $\{ (P_{y,i} < -\text{ref}) \text{ and } (P_{y,i+1} < -\text{ref}) \}$,

where $\text{ref} = 10$

(1) When $(|P_{y,i+1}| > |P_{y,i}|)$ and $(|P_{y,i}| > |P_{y,i-1}/\eta|)$, where $\eta=3$.

$$\text{Amplitude at current time } i, \Delta y = |P_{y,i+1}|$$

(2) When $(|P_{y,i+1}| > |P_{y,i}|)$ and $(|P_{y,i+1}| > |P_{y,i+2}/\eta|)$, where $\eta=3$.

$$\text{Amplitude at current time } i, \Delta y = |P_{y,i}|$$

(3) When $\{ (|P_{y,i+1} + P_{y,i}|/2) > |P_{y,i+1} - P_{y,i}| \}$

$$\text{Amplitude at current time } i, \Delta y = |P_{y,i+1} + P_{y,i}|/2$$

Otherwise

$$\text{Amplitude at current time } i, \Delta y = |P_{y,i+1} - P_{y,i}|$$

Duration-slope computation

$$\text{Duration-slope at current time } i, \Delta l = \frac{\sqrt{\Delta x^2 + \Delta y^2} \cdot \Delta y}{\Delta x}$$

The duration-slope is the product of the line length, $\sqrt{\Delta x^2 + \Delta y^2}$, with the line slope, $\Delta y / \Delta x$, at the point i .

3.3.3 Amplitude Normalization

Since the background level is neither stable nor consistent, the transitional segment containing a probable boundary needs to be normalized in every breath to detect points which cross the background level. However, as shown in Figure 3.8, the background level of the minimum-valued segment (*B* in the figure) is not always the same as the background level (*A*) near the probable boundary. In other words, duration-slopes in segment *A* can be considered as the burst activity level when duration-slopes of segment *B* are used as the background level. Thus, adaptive normalization is achieved by comparing the amplitude information from two segments in the background to alleviate the extent of a highly variable background level. Three levels will be defined, one belonging very probably to the background, one belonging very probably to the burst activity and a middle value threshold. The following method is used for the adaptive normalization.

In section 2.4, the minimum-valued segment from MTA81 and the coarse transitional segment were determined. These two segments play a key role in determining the background level. A graphical description of background level determination for the DEMG signal is shown in Figure 3.9.

The minimum-valued segment(121*0.5 msec) is selected from the original signal, instead of from the MTA81 signal. Peak-to-peak quantification and duration-slope computation for the segment are performed. The duration-slopes are sorted using a heapsort algorithm to reject the two lowest and the two highest-valued duration-slopes to reduce sensitivity to outliers in the background level. From the remaining duration-slopes, the average and the standard deviation value are computed. They are expressed as *base_mean* and *base_std* (*base_thld* as its threshold) for convenience. The background threshold of the minimum-valued segment is defined as:

$$\begin{aligned} \text{if } (base_std / base_mean) > 0.95 \quad & base_thld = base_mean + 1.5 * base_std \\ \text{Otherwise} \quad & base_thld = base_mean + 3.0 * base_std \end{aligned}$$

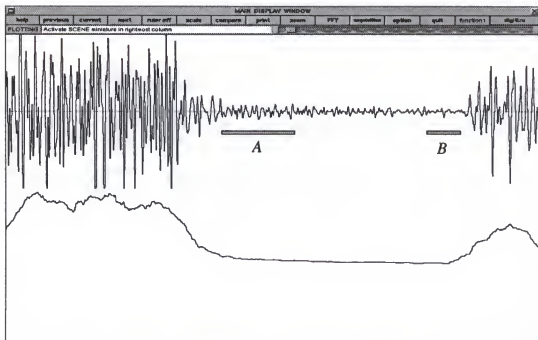


Figure 3.8. Example of unstable background level

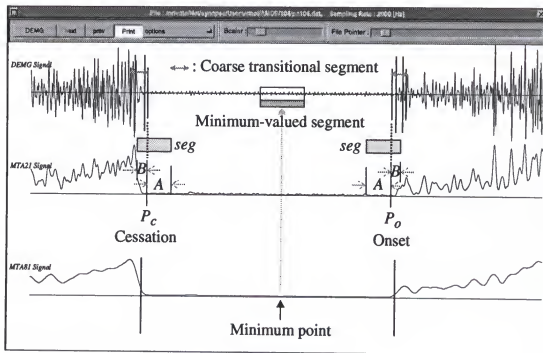


Figure 3.9. Background level determination based on coarse and minimum-valued segments

Another level which is related with post (or pre) burst activity is computed from an endpoint of the coarse transitional segment, *seg* in Figure 3.9, at point P_c for cessation (and P_o for onset). The segment length of *seg* is 90msec, such as $A=60\text{msec}$ and $B=30\text{msec}$. The peak-to-peak amplitude and duration-slope of *seg* are computed. The duration-slopes are sorted by amplitude and then the four lowest and the four highest-valued duration-slopes are rejected to reduce variation of the background level. From the remaining duration-slopes, the average and the standard deviation value are computed. They are expressed as l_mean and l_std (l_thld as a threshold). However, there is difficulty in selecting the length of A and B . Obviously, wide segment length is desirable because it gives more global characteristics of the background. Dependent upon the signal pattern in the transitional segment, the segment A may contain parts of the burst activity, and/or segment B may contain strong tonic activity of the background. Thus, further statistics are computed to derive a more stable average value. A fixed number of duration-slopes, i.e., 20, are selected from P_c to the right direction for cessation, and from P_o to the left direction for onset. Their statistics are also computed and called s_mean and s_std (s_thld as a threshold). The overall threshold for *seg*, called $base_a_mean$, $base_a_std$, and $base_a_thld$, is determined in four steps:

Step 1:

if $(l_std / l_mean) > 0.95$	$l_thld = l_mean + 1.5 * l_std$
else if $(s_std / s_mean) > 0.95$	$s_thld = s_mean + 1.5 * s_std$
Otherwise	$l_thld = l_mean + 3.0 * l_std$
	$s_thld = s_mean + 3.0 * s_std$

Step 2:

if $(l_thld / l_mean) \geq 0.95$	$base_a_mean = s_mean, base_a_std = s_std$
Otherwise	$base_a_mean = l_mean, base_a_std = l_std$

Step 3:

if $s_thld \geq (2.0 * l_thld)$	$base_a_mean = l_mean, base_a_std = l_std$
-----------------------------------	--------------------------------------------------

Otherwise $base_a_mean = s_mean, base_a_std = s_std$

Step 4:

if $(base_a_std / base_a_mean) \geq 0.95$

$base_a_thld = base_a_mean + 1.5 * base_a_std$

Otherwise $base_a_thld = base_a_mean + 3.0 * base_a_std$

The third threshold as a representative background level has been determined from two thresholds such as $base_thld$ of the minimum-valued segment which is usually located in the middle of the background and $base_a_thld$ of seg segment which is close to the boundary. The three final thresholds are determined and denoted as *Threshold1*, *Threshold2*, and *Threshold3*.

Step 1:

$temp_thld = 1.2 * base_thld$

if $temp_thld < 100.0$

$temp_thld = 2.5 * base_thld$

Otherwise

$temp_thld = base_thld$

Step 2:

if $base_a_thld < 0.8 * temp_thld$ $thld = 1.2 * base_a_thld$

else if $(0.8 * temp_thld) \leq base_a_thld < 1.2 * temp_thld$

$thld = temp_thld$

else if $(1.2 * temp_thld) \leq base_a_thld < 2.5 * temp_thld$

$thld = 1.2 * temp_thld$

else if $(2.5 * temp_thld) \leq base_a_thld < 5.0 * temp_thld$

$thld = 1.2 * temp_thld$

Otherwise

$thld = temp_thld$

Step 3:

$Threshold1 = thld$

$Threshold2 = 2.0 * thld$

$Threshold3 = 3.0 * thld$

All the criteria used in thresholding have been empirically determined by examining a large number of background and transitional segments.

3.3.4 Quantization of Duration-Slope

After the peak-to-peak detection and duration-slope computation around the boundary candidate, 20 duration-slopes, 10 before and 10 after the boundary candidate, are collected and compared with the background thresholds described in section 3.3.3. The input pattern of the neural network consists of the quantization levels.

Experimentation revealed that the duration-slope greater than *Threshold2* belongs to the burst activity group. Most duration-slopes which may or may not belong to the background are in the range of *Threshold1* and *Threshold2*. Thus, it is necessary to divide this range into smaller levels in order to increase boundary detection accuracy. Thus, the normalized amplitude range is split into four subthresholds. The amplitude between subthresholds is called *margin*. Following are the procedures for determining quantization levels: A pictorial description of the final background thresholds and associated quantization levels is shown in Figure 3.10.

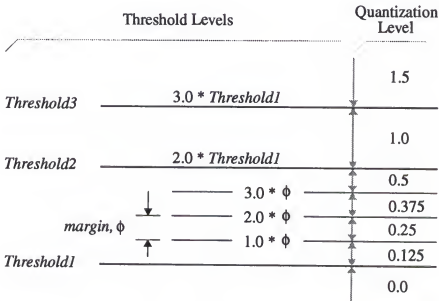


Figure 3.10. Description of threshold levels and corresponding quantization levels

$$\text{margin} = (\text{Threshold2} - \text{Threshold1}) / 4.0$$

if $\text{duration-slope} \leq \text{Threshold1}$ quantization level = 0

else if $\text{Threshold1} \leq \text{duration-slope} < (\text{Threshold1} + \text{margin})$

quantization level = 0.125

else if $(\text{Threshold1} + \text{margin}) \leq \text{duration-slope} < (\text{Threshold1} + 2.0 * \text{margin})$

quantization level = 0.25

else if $(\text{Threshold1} + 2.0 * \text{margin}) \leq \text{duration-slope} < (\text{Threshold1} + 3.0 * \text{margin})$

quantization level = 0.375

else if $(\text{Threshold1} + 3.0 * \text{margin}) \leq \text{duration-slope} < \text{Threshold2}$

quantization level = 0.5

else if $\text{Threshold2} \leq \text{duration-slope} < \text{Threshold3}$ quantization level = 1.0

Otherwise quantization level = 1.5

During quantization, the duration-slopes of the line-segments which are beyond the limits of the coarse transitional segment are set to zero because all boundaries are inside the coarse transitional segment. Thus, by setting the high quantization level for sparsely occurring burst activity in the background to zero, more stable input patterns to the neural network can be provided.

3.4 Factors in Implementing the Neural Network

This section discusses some factors to be considered when the neural network is implemented such as the number and type of input pattern, the collection of training data and learning, and network architecture. All these factors are determined according to the experimentation for the given training data sets. The number of nodes in the input layer is first decided, and then input level and network architecture are determined.

3.4.1 Number of Input Pattern

Transformed data from the raw EMG signal are used as input patterns to the network. Since the resolution of boundary detection was assumed a line segment, the modified line length (duration-slope of half-wave) between peaks neighboring the boundary is used as the basic unit of the input vector. For the network to come up with diverse characteristics of the transitional signal patterns, 20 basic units corresponding to roughly 40 msec seem to be appropriate. This number has been somewhat verified (see section 3.4.4) to be practical for the EMG boundary classification in term of the system training and classification result. The number of the input pattern is primarily concerned with the determination of other factors.

3.4.2 Input Quantization

Although the raw signal is transformed into normalized line segments using peak detection, they have a large dynamic range (12 bits), resulting in a wide range of input levels which made the network convergence very difficult. It had been shown that a small number of discrete input levels makes the system adaptation stable and fast. We experimentally determined the best quantization step. The network was tested using the same data sets with different numbers of quantization levels for a given architecture (20 nodes in an input layer, 10 nodes in a hidden layer, and 1 node in an output layer). The test result using 104 test cases, momentum of 0.9, learning rate of 0.7, maximum total error of 0.0001, maximum individual error of 0.00001, and maximum number of iterations of 500, is shown in Table 3.2. The input dynamic range for the test cases are as follows: Case 1 = 0.0, 0.5, 1.0, Case 2 = 0.0, 0.5, 1.0, 1.5, Case 3 = 0.0, 0.5, 1.0, 2.0, and Case 4 = 0.0, 0.5, 1.0, 2.0, 3.0. Based on the result, Case 2 which has 4 uniformly distributed levels is selected because it has the minimum system error as well as it trains in the minimum number of iterations.

The result shows the general idea about the number of discrete input levels and its level to be used in the system. Because the normalized amplitude of probable boundaries

has been shown experimentally in the level of 0.0 through 0.5, the smaller partition can improve the accuracy of boundary detection in time. Thus, the input level which is less than 0.5 is actually divided into four smaller levels in the segmentation system, i.e., 0.0, 0.125, 0.25, and 0.375 (refer to Figure 3.10).

Table 3.2. Result of four cases for different input quantization

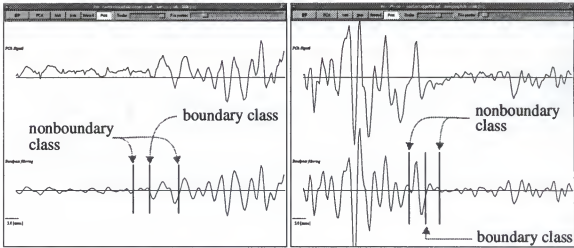
Cases Output Summary	Case 1		Case 2		Case 3		Case 4	
	0	1	0	1	0	1	0	1
mean	0.002189	0.985724	0.002008	0.992271	0.002691	0.980063	0.036609	0.925919
standard deviation	0.00744	0.017373	0.006188	0.011938	0.00864	0.007209	0.176717	0.000239
minimum	0	0.928984	0	0.949052	0	0.947152	0	0.925021
maximum	0.04047	0.998301	0.048409	0.998043	0.056322	0.988847	0.925997	0.925997
number of iterations	55		54		168		500	
system error	9.8E-05		5.6E-05		9.5E-05		0.013654	

3.4.3 Training Sets

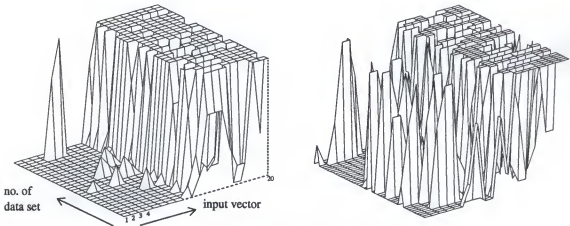
The training sets are collected from the raw signal which shows clear boundaries. Several training sets can be chosen from a transitional segment. The training sets from onset or cessation transitional segments are treated in the same way because, by reversing the signal, those from the cessation segment are indistinguishable from the onset. During the training of the neural net, the desired output is set to 1 if the input belongs to a boundary class, and otherwise it is set to 0, i.e., a non-boundary class. The network weights are then adjusted to minimize the total system error between the network outputs and the desired outputs.

The difficulty in collecting training sets is that although the training sets are obtained from clear boundaries in the EMG signals, there are still lots of possibilities of selecting non-boundary examples. In this research, the selection of the non-boundary class is any line segment which is away from the boundary by at least two line-segments. The criterion for non-boundary class is to collect the points which are in the clear background or the clear burst activity. Examples of the selection of training data set and input patterns of discrete levels are shown in Figure 3.11. The total number of training sets is 192.

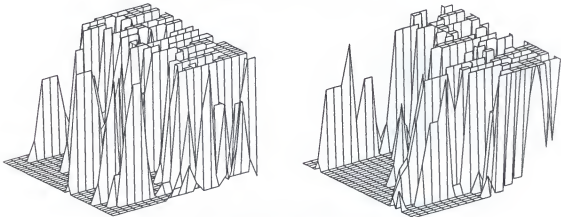
Note that the sigmoid, because of its sharp transition, already plays an important role in focusing network "attention" during training on data samples that are not clear. In my opinion, for good discriminant function between classes, more training examples which are near the separating discriminant line between the boundary class and the non-boundary class are preferred.



(a) Examples of boundaries selection



(b) Data set of boundary class



(c) Data set of non-boundary class

Figure 3.11. Example of training data set

3.4.4 Network Architecture

Several factors have been already determined such as 20 input nodes in the input layer, 1 output node, and signal quantization. The remaining is to determine the number of hidden layers and hidden nodes. Generally, the decision regions required by any classification algorithm can be generated by a three-layer feedforward network [Lia87] because it can generate arbitrarily complex decision functions. It is important to note that the first layer serves a special function that is quite different in nature from other layers. Once the first layer partitions the input space into cells, the other layers can only group these cells together to form decision regions.

To determine the appropriate network topology for the discriminant function, the networks of different structures were tested using already chosen training data sets. Table 3.3 shows the result in which 192 training data sets, maximum system error of 0.0001, maximum individual error of 0.00001, and maximum iteration number of 2000 are used. In the table, a means the number of hidden layers, b and c mean the number of nodes of the first and second hidden layers. For example, (2, 5, 2) shows that the number of nodes of the first and the second hidden layer is 5 and 2, respectively in 2 hidden layers. As shown in the table, there are two clusters (area A and area B) of small numbers of iterations that produced small system error. The detailed information for those areas is shown on the right side of the table. The best architecture is selected in the first shaded area because it has less complex architecture and requires less computation time for training. Thus, the topology of one hidden layer with 18 nodes is chosen.

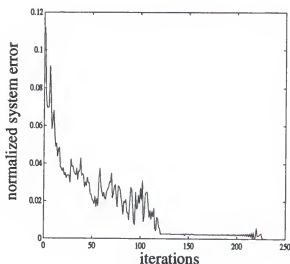
The learning curve (total system error versus the number of iterations) for the selected topology is presented in Figure 3.12. Part (a) in the figure is drawn on a linear scale and (b) on a logarithm scale. In the logarithm scale, two abrupt reductions of the system error which cannot be easily recognized in the linear scale are shown. The threshold of the system error for training is set at the second transition, i.e., 0.00005. In the automated segmentation system, the training is required only one time before signal analysis.

Table 3.3. Experimental result for determination of network architecture

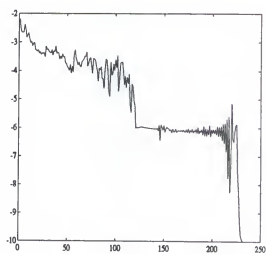
iteration number	system error	architecture (a, b, c)
2000	0.019903	(1, 3)
2000	0.012232	(1, 5)
259	0.000100	(1, 7)
2000	0.007819	(1, 9)
2000	0.002611	(1, 11)
A	398	0.000099 (1, 13)
	407	0.000097 (1, 15)
	259	0.000079 (1, 17)
	271	0.010079 (1, 19)
2000	0.035037	(2, 5, 2)
2000	0.017446	(2, 5, 4)
2000	0.015604	(2, 8, 13)
2000	0.012740	(2, 8, 15)
2000	0.002561	(2, 11, 2)
2000	0.010791	(2, 11, 5)
2000	0.052103	(2, 11, 8)
2000	0.007673	(2, 14, 3)
432	0.000097	(2, 14, 5)
2000	0.002604	(2, 14, 8)
435	0.000039	(2, 14, 11)
2000	0.070661	(2, 17, 3)
B	284	0.000089 (2, 17, 5)
	356	0.000091 (2, 17, 8)
	272	0.000080 (2, 17, 11)
	302	0.000051 (2, 17, 14)
2000	0.115929	(3, 5, 3, 2)
2000	0.007586	(3, 8, 5, 2)
2000	0.002574	(3, 17, 15, 7)

iteration number	system error	architecture (a, b, c)
398	0.000099	(1, 13)
348	0.000072	(1, 14)
407	0.000097	(1, 15)
299	0.000100	(1, 16)
259	0.000079	(1, 17)
193	0.000099	(1, 18)
271	0.000079	(1, 19)
262	0.000100	(1, 21)

iteration number	system error	architecture (a, b, c)
1564	0.000054	(2, 17, 6)
1416	0.000084	(2, 17, 7)
356	0.000091	(2, 17, 8)
2000	0.010382	(2, 17, 9)
326	0.000052	(2, 17, 10)
272	0.000080	(2, 17, 11)
704	0.000067	(2, 17, 12)
361	0.000072	(2, 17, 13)
302	0.000051	(2, 17, 14)
348	0.000041	(2, 17, 15)
259	0.000036	(2, 17, 16)
531	0.000096	(2, 17, 17)



(a) linear scale



(b) logarithm scale

Figure 3.12. Training curve for the selected topology

3.5 Determination of the Final Boundary

The final boundary can be considered representative for the given transitional segment and is determined based on the maximum *a posteriori* boundary estimate. It is defined as the first point at which the half-wave with the highest probability exceeds the background level. The background level for the final boundary is defined as an average value of half-waves which are detected from the background endpoint (f_f or f_b in Figure 3.13) of the fine transitional segment (A in the figure), that is, the nearest background point to the boundary estimates.

Figure 3.13 shows an example of several boundary estimates for onset and cessation, and depicts the final boundary detection. The lines f_f and f_b mean onset and cessation endpoints of the fine transitional segments. This figure shows three boundary estimates of onset, p_{f1} , p_{f2} and p_{f3} , and three boundary estimates of cessation, p_{b1} , p_{b2} and p_{b3} . Here, the final boundary decision for onset is described.

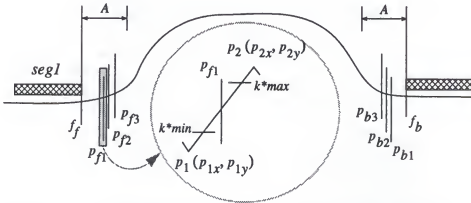


Figure 3.13. Representation of boundary estimates and their notations for explanation of the final boundary detection

The endpoints, p_1 and p_2 , of the maximum posterior line segment are detected. For a point p_i , p_{ix} and p_{iy} , refer time and amplitude at time i . All peaks are detected in the segment of *seg1* in the figure, and fifteen peaks before f_f (or after f_b) are chosen. After sorting by amplitude and rejecting the two largest and smallest peaks, the minimum level,

min, is defined as the average of the four smallest peaks and the maximum level, *max*, is defined as the average of the four largest peaks. The *min* and *max* value are used as the background level for the final boundary detection.

The decision rules for “positive” slope of the line segment which contains p_{fi} are: (Graphical description is shown in Figure 3.14.)

if ($p_{1y} < (k*min)$) (refer Figure 3.14(a))	final boundary = p_{1x}
else	
if ($p_{0x} < p_{1x}$) (refer Figure 3.14(b))	
if ($p_{2y} > (k*max)$)	final boundary = p_{mx}
else	final boundary = p_{2x}
else (refer Figure 3.14(c))	final boundary = p_{0x}

The decision rules for “negative” slope of the line segment which contains p_{fi} are:

if ($p_{1y} > (k*max)$) (refer Figure 3.14(d))	final boundary = p_{1x}
else	
if ($p_{0x} < p_{1x}$) (refer Figure 3.14(e))	
if ($p_{2y} < (k*min)$)	final boundary = p_{mx}
else	final boundary = p_{2x}
else (refer Figure 3.14(f))	final boundary = p_{0x}

where $k=3.0$.

The above procedure can also be applied to the final boundary determination of the cessation.

Since the half wave in the burst activity consists normally of about 4 points (=2msec), the final boundary procedure has only a modest resolution improvement over an arbitrary selection of points within the half wave.

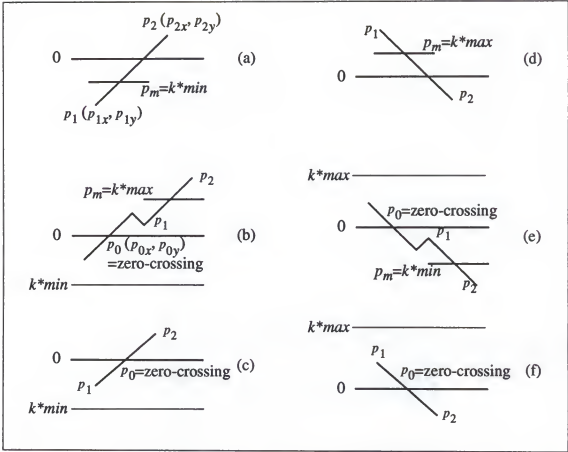


Figure 3.14. Graphical description for final boundary decision

CHAPTER 4

SIGNAL PROCESSING AND VISUALIZATION SYSTEM

Two sets of programs were developed in this dissertation: First, a general-purpose interactive signal processing environment called "*Signal Editor*" was designed for displaying and quantifying signals. The tool has been developed to visualize multiple channels of data easily, to observe the effect of digital signal processing algorithms on the data simultaneously, and to evaluate signal algorithms for any type of signals quickly. Second, an automated segmentation system was developed to implement the segmentation model for the respiratory EMG signals which has been discussed in previous chapters.

Both programs were designed in the NeXT workstation using an object-oriented programming paradigm and the segmentation module can be totally integrated in the Signal Editor. The general software structure will be described and many of the system features will be illustrated with examples. All the figures displaying EMG signals in this dissertation were obtained using these two programs.

4.1 Introduction

Many engineering problems require the development and validation of algorithms such as filtering, detection of specific features or events which are visualized or defined in the time domain. The Signal Editor helps the user evaluate such algorithms in two ways: signal measurement for accurate evaluation, and graphic visualization for qualitatively examining results. User-developed signal processing functions are easily imported into the Signal Editor's program structure. The user can navigate to any position in a file, which may have multiple channels of data, and select the appropriate data segment for testing his/her algorithms.

Thus, this window-based and mouse-driven software package serves as a prototype for the design, analysis, validation and evaluation of user-defined algorithms. It also has several features for visualizing multiple data channels for qualitative analysis.

Another motivation for developing the Signal Editor is the fact that *"Computers were developed to work with text - as substitute for a typewriter - but that is not the correct paradigm for biomedical engineering, which lives with images and signals. To make the computer work optimally for us, we need to develop tools to function like a text editor, but for graphics and signals instead of for letters and words."* [Ne90]. The Signal Editor can be used to visualize all types of electrical signals - from EEGs to EMGs to audio waves or music signals. The user can access a previously collected signal file, observe temporal and spatial characteristics of the signal by displaying both single and multichannel data on the screen.

Working with the Signal Editor, the computer functions as a scope, or actually, more like a microscope, since it can zoom in on specific portions or events and offer very accurate time and amplitude measurements of signals. Moreover, we can collect specific events or activities into files for various purposes. For example, a biosignal-analyst can collect the signal segments of clinically significant features by searching the signal. These features give them more detail and concrete characteristics about signals. Therefore, the Signal Editor can assist the bio-engineer in designing the signal detection algorithms as an initial stage to develop a computer-based detection system, as well as the physiologist in the visual clinical assessment of multichannel biosignal recordings. It may also serve as a paperless workstation which replaces the multichannel polygraph now being used for biosignal monitoring.

The characteristics of the system are those of generality, expandability, and flexibility. New algorithms can be integrated by following a standard format and can be easily evaluated in the Signal Editor. The algorithms in the system's function library can be updated (addition of new functions and deletion of old functions). While the program is

running, the combination of algorithms can be changed at any time so that the functions to be evaluated are interactively and dynamically tested. The user can control the Signal Editor by using the mouse exclusively. All the information is presented to the user in the form of a window environment with mouse sensitive command buttons.

Binary input data can be collected in two ways: (1) The signals can be transferred from the personal computer to the NeXT computer through the Ethernet network (off-line). (2) They can be digitized real time using the Dazzl analog-to-digital converter interfaced to the NeXT bus.

The window-based automatic segmentation system is also designed so that the user can control the flow of signal processing and access any windows to check intermediate results visually. The main software modules are associated with the data handling, signal processing, and display. There are five major windows which correspond to noise rejection, signal segmentation, detection of boundary candidates, boundary editing and boundary classification (training data collection and neural network learning), and boundary estimation. The segmentation model is evaluated based on results (transitional segments and the final boundary) of the segmentation system. All results can also be examined visually through the display window.

4.2 Environment of System Development

Both systems run under Mach on NeXT computer (both NeXTcube and NeXTstation) and all the software is written in Objective-C, an extension to C that adds object-oriented concepts to the language. There are four levels of software between a NeXT application program and the hardware that executes it: NeXT Interface Builder (IB), Application kit or Objective-C software "kits", NeXT Window Server, and Mach operating system. The hierarchical structure of the NeXT software is shown in Figure 4.1.

IB is a powerful tool that lets the user graphically design application interfaces such as menu, button, window, and slider. Various items and windows provided by IB simplify

building user-friendly, interactive, graphic, menu-driven interfaces. It also makes it easy to establish connections between user-interface objects and its corresponding code in order to create a programming structure for each new object. The software kits define a number of classes, or object templates, that we can use in applications. So every application uses an Application Kit to implement the NeXT window-based user interface. The NeXT Window Server is a low-level background process used by the Application Kit to manage windows and to send user events, such as mouse and keyboard actions, back to an application. The Window Server includes a Display PostScript (DPS) interpreter that is used for all drawings of text and graphics on the screen or printed page. Mach is a multitasking operating system, which acts as an interface between the upper levels of software and other processors (central, floating-point, and digital signal processors) in the NeXT computer.

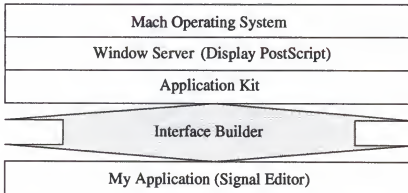


Figure 4.1 NeXT software hierarchical structure for applications development

The implementation of both systems is mainly based on the IB and the Application Kit. After designing all interface objects in the IB, the programming codes of functional routines are written in implementation files which are also generated by the IB. Sometimes new objects are generated to implement new specific behavior. The full resolution (1120 * 832 pixels) of the NeXT's PostScript screen can be used for waveform display. The detailed description of four software systems and the concept of the object-oriented programming structure is as follows:

Interface builder. IB is an extremely useful environment for designing a system. It displays a window that represents the application's user interface and provides graphic access to a number of standard interface objects, such as text fields, buttons, and menus. To design an interface, we simply drag the interface objects into the application's interface window and place them where we want them. We can inspect and modify an object to create a particular look. Interface objects understand user events such as mouse and keyboard actions and automatically perform basic display operations when an action is directed at them. For example, a button knows how to highlight itself graphically when the user clicks it, a text field displays the characters that the user types in it, and a window disappears when its close button is clicked.

In addition, IB has tools for connecting interface objects to each other. For instance, we can connect a button to a panel so that the panel comes to the front when the user clicks the button, or we can connect a slider to a text field so that the value displayed in the text field is continuously updated as the user drags the slider's knob up and down. NeXT provides the code for the basic interface operations; we can also design our own objects and actions and connect them in IB. Most applications require more work than simply defining an interface. The other facet of IB, its creation of a programming environment, makes it a good place to start a new project even if the interface is trivial compared to the amount of programming the project requires. IB can automatically create a UNIX makefile (the script for the application's compilation routine), some basic source codes, and the header files that the application needs to compile.

The application kit. All applications use the Application Kit regardless of their purpose or complexity. The buttons, sliders, and windows that we use to design an interface with IB are defined as classes in the Application Kit. Also, it is through this Kit that the application is able to draw on the screen and receive events from the user. Objective-C and the software kits make it easy to create our own class of object. One of the features of the language is that it supports class inheritance; this means that we can create a class that

inherits the attributes of another class. For example, we can create another button class that inherits the attributes of the Application Kits's button class. Versions of buttons will be able to do everything that the Kit version can do. In addition, we can add to it the specialized functionality that the application requires.

NeXT window server. The Next Window Server is a low-level background process that creates and manipulates windows on the screen. The application establishes a connection with the Window Server through the Application Kit and opens one or more windows. Windows provide a vehicle for communication between the user and the application. The Window Server manages this communication as it fulfills two functions: it draws images on the screen according to instructions sent from the application, and it sends user events, such as mouse and keyboard actions, back to the application.

The Window Server draws and prints images with NeXT's implementation of the Display PostScript (DPS) system. DPS provides an interactive, display-oriented environment that is independent of any window system. All the DPS operators and the NeXT extensions to the language can be accessed as C functions. Besides drawing images on the screen, the Window Server also identifies user events and dispatches them to the application. Through a mechanism defined in the Application Kit, the event is forwarded to the appropriate object.

Mach Operating System. The Mach operating system, multitasking operating system, provides a system of interprocess communication, a large virtual memory space, memory-mapped files, and multiple threads of execution within a single address space. Mach gives programmers the entire standard UNIX environment. Every running application is a separate process. In Mach, several processes may be running concurrently. For example, the Window Server process runs at the same time as all currently executing application processes.

Object-oriented Programming Structure. The programming style of the NeXT computer is exclusively object-oriented. Objective programming gives many advantages [Cla90]. It is easier to debug, to maintain, and to enhance. As programs become large and complex, the objective programming is more effective than a conventional program structure. A feature in object programming is encapsulation. Functions, procedures, and data are rolled together into objects. Applications are made up of interacting objects, with each object having a specific functionality. Objects may have instance variables which contain data. The instance variables are private data. The procedures and functions used by objects are called methods, which are also specific to a particular object. Once an object is created, we only need to remember what the object does and what methods are used by it without worrying about what is happening inside the object. All the functionality of the object, data, procedures, and functions is encapsulated within the object.

Object programming groups objects into classes, which can be arranged in a hierarchy. Every class is a subclass of another class, and every class has a super class that is one step higher in the hierarchy. Each class shares the instance variables of its superclass which eliminates much duplication of the program code. In the NeXT computer, everything begins with the "Object" class. It is a primary class, i.e., the origin of all objects which is defined in NeXT system. Other classes are always defined relative to the Object class, which is the only class without a superclass. Each subclass automatically inherits the instance variables and the methods of its superclass, and that class's superclass, and so on, all the way up the line of Object. Each subclass defines additional methods and variables that are used to create more specific behavior. If the classes are properly thought-out and written, there is very little duplication of programming code. In most cases, the classes and objects in NeXT's Application Kit can be used. If new behavior wants to be added, a programmer can add new instance variables, new methods, or variables to the existing methods. An example of class hierarchy for windows, buttons, sliders, and menus used in the Signal Editor and the automated segmentation system is shown in Figure 4.2.

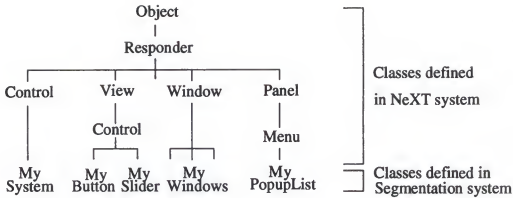


Figure 4.2. Hierarchical structure of classes used in the system

Object-C language used in the NeXT is a superset of the ANCI C language. All the conventional features of C are provided in addition to the new programming features. Objects communicate with other objects by sending messages whose statements contain the objects's name, the method, and method arguments.

To get an object to do something, a programmer must send it a *message* telling it to apply a method. In Objective-C, *message expressions* are enclosed in square brackets: [*receiver message*]. The receiver is an object, and the message tells it what to do. In program code, the message is simply the name of a method and any arguments that are passed to it. When a message is sent, the run-time system selects the appropriate method from the receiver's repertoire and has the receiver apply it. For this reason, the method name in a message is called method selector. Messages in Objective-C can appear in the same syntactic positions as function calls in a standard C. However, since methods belong to an object, messages behave differently from function calls in the following ways: an object has access only to the methods that it can perform, and it can't confuse them with the methods of another object, even if the other object has a method with the same name. This means that two objects can respond differently to the same message and perform in a unique way. This feature, which plays a significant role in the design of object oriented programs, is sometimes referred to as polymorphism. A method has automatic access to all the receiving object's instance variables.

A crucial difference between function calls and messages is that a function and its arguments are bound together in the compiled code, but a method and a receiver object are not united until the program is running and the message is sent. A method is called through a run-time messaging routine that locates the method named by the selector and passes the object's instance variables to it. This dynamic binding of methods and receivers works hand-in-hand with polymorphism to give object-oriented programming much of its flexibility and power.

4.3 General Signal Algorithm Processing and Visualization Tool - "Signal Editor"

The Signal Editor can display both the original and processed signals of any combination of algorithms for visual validation. It zooms in specific portions of the signal to make amplitude/duration and slope measurements for precise evaluation of an algorithm. It is also possible for the user to develop his/her own functions for special measurements. Therefore, the user can easily test his own functions visually or by measuring parameters. In addition, the Signal Editor creates files with selected portions of the signal, creates printouts, and produces power spectral estimations of selected portions. These facilities reduce the development time of the algorithms in terms of data searching and manipulation to handle large data files, measurement of parameters, and graphical displays of original and processed signals for the validation of the algorithm. The algorithms which had been tested can be registered into a built-in function library for later use.

This software package is frequently useful for visually comparing original signals with the results of signal processing algorithms on single and multiple channel data. The Signal Editor allows two signals from any combination of algorithms to be compared. For instance, the user may need to distinguish differences among algorithms (e.g., digital filters with different filter coefficients or parameters) or compare different algorithms (e.g., FIR and IIR, or median and peak-detection) for the same signals. Any filtering done on the data to enhance the signal features may also distort the temporal characteristics of the signal of

interest. The filtering effects are visually observed by displaying the data before and after filtering. Therefore, it offers an efficient and interactive way of visually comparing signal processing algorithms. It is obvious that the visual feedback considerably decreases the algorithm development time.

4.3.1 Overall System Architecture

The system of mouse-driven software is designed so that the user can access and control the different aspects of signal visualization and analysis. The main software modules are associated with the data search, data processing, manipulation of functions, and display. Figure 4.3 shows the overall functional block diagram of the Signal Editor. Versatility in signal analysis and visualization can be achieved by software modularization with the following data structure: data in mass storage are not directly displayed on the monitor. Data Handling Manager reads fixed length data segments from the file according to the user's selection and then loads them into a Global Input Buffer (GIB). The data in a Global Output Buffer (GOB) are displayed on the Main Display window and/or one of subwindows by the Display Manager, which changes amplitude and time scales, zooms in on a specific event or segment, and etc.

The Function Executer can use dynamic buffers for temporal storage of results during function execution and modify the buffer at the user's command about the combination of functions, and then it is sent to the GOB for display. The temporal buffer receives data from the GIB and sends it to an algorithm. Processed data are saved in another intermediate buffer that can be mapped onto the GOB and/or can serve as the input to other signal processing modules. The signal processing units can be applied sequentially based on the combination of algorithms. With this flexible arrangement, the user can test and evaluate visually single or multiple signal processing units. When there are no functions (direct display), GIB data is sent to GOB without generating dynamic memories as intermediate buffers. This sequence (Figure 4.4) is repeated for every display channel.

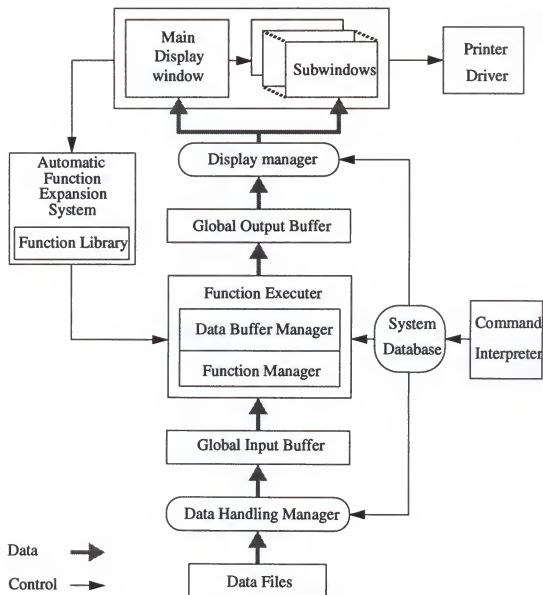


Figure 4.3. Functional block diagram of Signal Editor

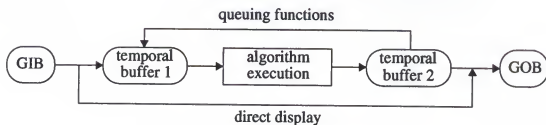


Figure 4.4. Data flow for queuing functions

Modifications of the buffer are performed by applying the digital signal processing algorithms kept in the Function Library. The user can add or delete his own functions, provided the set of standard procedures is followed. The Function Library, which contains template signal processing algorithms (such as FIR, IIR, waveform detection, and etc.), can have up to 16 main function units. Each main function unit can include up to 16 subfunction units (such as FIR1, FIR2, ... in FIR main function unit), which may have different values of parameters or coefficients. Thus, the total number of algorithms to be implemented in the system is 256.

The user specifies the algorithm discriminator, user-defined symbol, in the registration of subfunction units to distinguish them (e.g., FIR1 as a lowpass FIR filter with a 10Hz cutoff frequency, FIR2 as 20Hz, and etc.). The maximum number of functions to be processed in one channel is currently 5. If the user wants to increase the number of functions, he/she can redirect the output of a channel to the input of another channel (refer to Figure 4.5). Thus, the maximum number of subfunctions to analyze the signal is 35, based on the algorithm arrangement of the display channel. The subfunctional signal processing units are connected and performed in cascade.

The configuration of the Signal Editor is kept in the System Database, including the currently active data file name, file information, time scale information, channel number for display, formation of GIBs and GOBs, and combination of the function units. The user interacts with the System Database through the main system menu in Setup window of Figure 4.5. While the program is running, the user can access these menus, at any point or time, to change any configurations for the diverse signal analysis. This interactive operation provides the user with more flexible capability to evaluate algorithms easily and dynamically and to visualize the processed results. Figure 4.5 exemplifies a session to set up the System Database, in which the user selected the data file `"/private/Net/synapse/Users/choi/program/EEG.dat"` with information of the data file such as channel number, sampling rate, number of data bytes, and data swap status.

SETUP WINDOW

SIGNAL EDITOR

Version 1.0
UF CNEI Lab.

Instructions: The time duration a frame is 524.000000 msec

File Descriptions

File Name:

File Ch no.: Sampling Rate:

Data Bytes: Data Swap:

Total Display Ch no.:

Compression Ratio:

FUNCTION ASSIGNMENT of DISPLAY CHANNEL

Mode Selection

MAIN FUNCTION SELECTION

Median	Peak	PWR	Derive
Mean	Shift	Gauss	MTA
THR	POINT	WIN	FIR

SUBFUNCTION SELECTION

fir1	fir2	fir3	fir4
fir5			

Display of Channel Function

Ch 1	in_buf1	outbuf1				
Ch 2	in_buf1	peak1	outbuf2			
Ch 3	outbuf2	mean5	fir1	outbuf3		
Ch 4	in_buf6	fir2	outbuf4			
Ch 5						
Ch 6						
Ch 7						

Figure 4.5. Setup window of Signal Editor

The Signal Editor can manipulate a binary data file of 1 byte or 2 bytes. However, data handling in the NeXT computer may be different from other computers. For this case, especially in 2 bytes data, the user has to set the swap status flag in order to switch the upper byte with the lower byte before signal processing. When the data file is read by selecting the file in File Open window shown in Figure 4.15(a), the system checks both the data file of extension “-.dat” and its header file of extension “-.h” which keeps information of the data file.

The user can change the display rate to navigate on the data file. The display rate tells how many data points in a single channel are displayed on screen. The current

available range of the compression rate is 0.1 to 10.0 with step size of 0.1, which means that minimum data points are 0.1×1048 (maximum pixel number on display window) and maximum points are 10.0×1048 points. The user can also select the total number of channels for display, so display configuration should be arranged at the Display of Channel Function box. As shown in Figure 4.5, for example, the first display channel (*outbuf1*) of the Main Display window will directly display the data of the first channel (*in_buf1*) of the data file. The second channel (*outbuf2*) will display the output of the peak-detection algorithm using the first channel data (*in_buf1*) as input. The output of the second channel (*outbuf2*) is redirected to the input of the third display channel; thus the third channel displays the result of the processing of two algorithms (mean and lowpass filter). The fourth channel displays the output of lowpass filter (*fir2*) for input data of the sixth channel (*in_buf6*).

The arrangement of input and output buffers and built-in functions can be chosen by clicking the boxes in Mode Selection column. There are two modes: Function and Data Buffer mode. Data Buffer mode shows all *in_buf* and *outbuf* boxes for the GIB and GOB, and Function Mode shows the main functions and their subfunctions registered in the Function Library. The user can select the subfunctions and input/output buffers by clicking appropriate boxes and can move those selections into any box locations in the Display of Channel Function also by clicking the corresponding box to assign functions and input/output buffers of each display channel.

The Command Interpreter accepts the command information in the System Database, interprets, and distributes it to appropriate modules. The Signal Editor can also digitize a signal by clicking the Digitization box. The Dazzl (Model 16/12) analog-to-digital converter which is controlled by software commands, provides data acquisition for 16 single-ended or 8 differential channels with 12-bit resolution and has a maximum sampling rate of 200 kHz with real time. The digitized data sets are also directly brought into the Signal Editor, displayed in the Main Display window at the same time, and saved

into user-specified files in binary format. The digitization can be performed only in the NeXTcube because of the board installation into the NeXT bus slots.

Another important facility is a laser printer driver, which produces permanent archives of the computer screen. Whenever the print button is clicked, the Signal Editor sends the user a message to invoke another program which is called "Scene". The Scene is a general software for capturing a whole screen or a specific window, processing an image (i.e., cut, copy, paste, etc.), saving, and printing the captured image. However, the only weak point is its low resolution. The Scene window is shown in Figure 4.15(d).

The user can also register new functions by clicking the Function Update box. The built-in Function Library is expanded by executing the Automatic Function Expansion System (AFES). The AFES can add new functions and delete old functions with their subfunctions from the Function Library. All function information processed in the AFES is stored in the System Database so that the Function Executer sequentially reads the names of function objects and executes the associated routines.

4.3.2 Hierarchical Structure of Windows

An object-oriented program is typically built from a variety of objects. Programs often use more than one object of the same type of class such as windows. In objective-C, we define objects by defining their class. The class definition is a prototype for a kind of object; it declares the instance variables that become part of every object belonging to the class and defines a set of methods that all objects in the class can use. The hierarchical structure of window classes in the Signal Editor is illustrated in Figure 4.6. The "Window" in the figure means the window class defined in the NeXT Application Kit, which contains a variety of classes for objects that draw on the screen, and responds to user actions on the keyboard and mouse. All window classes in the Signal Editor are linked together in a hierarchical tree under the Window class.

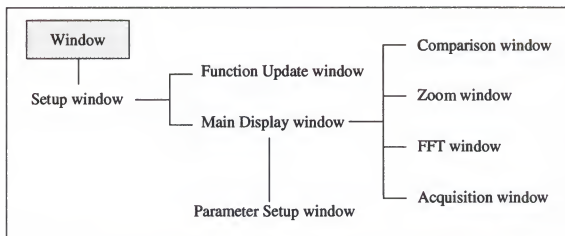


Figure 4.6. Hierarchical structure of windows in Signal Editor

Every class has either a superclass or subclass, or both. When we define a new window class, we must link it to the hierarchy by declaring its superclass. Each class inherits both instance variables and methods from its superclass. When a class object creates a new instance, the new object contains not only the instance variables that were defined for its class, but also the instance variables defined for its superclass, and for its superclass's superclass. This is same for the class's methods which tell an object to apply a method or function. Any new class defined in the Signal Editor can, therefore, make use of methods defined for all the classes above it in the hierarchy. This type of inheritance is a major benefit of the object-oriented programming.

The user command sequences, especially in accessing windows, also follow the window hierarchy so that, for example, the commands in the Zoom subwindow are performed after opening data files in the Setup window and displaying channels of signals in the Main Display window because all subwindows can be initiated by inheriting instance variables and methods defined in the Setup window and/or in the Main Display window such as compression rate, sampling rate, input and output buffers, and etc.

4.3.3 Functional Description and Visualization

The identified tasks for signal visualization and analysis are implemented into program environments. For easy interfacing, each environment corresponds to a window where the user selects the available commands in command buttons. There are six windows excluding the Setup window: Main Display window, Comparison window, Zoom window, FFT window, Acquisition window, and Function Update window. Signal measurement in the Zoom window, registration of new functions in the Function Update window, and visualization facilities and some utilities in other windows are described.

Main display window. Signal browsing is accomplished in the Main Display window shown in Figure 4.7. The maximum number of display channels is 7. The available commands are displayed on the screen's top row as buttons. The user simply selects the required function by clicking a button of the system. The window displays the processed outputs of the Function Executer for each channel according to the frame size. The frame size means the length of time the signals are drawn in a display channel on screen, and is given by:

$$\begin{aligned} \text{Minimum frame size} &= \text{minimum compression rate} (=0.1) * \text{sampling period} (= 1/ \\ &\quad \text{sampling frequency}) * \text{screen width on display window} \\ &\quad (=1048 \text{ pixels}) \end{aligned}$$

$$\begin{aligned} \text{Maximum frame size} &= \text{maximum compression rate} (=10.0) * \text{sampling period} * \\ &\quad \text{screen width} (=1048 \text{ pixels}) \end{aligned}$$

For the compression rate of 1.0, all the original data are drawn on the screen since the size of the input buffer is the same as that of the screen width. The display strategy for less than 1.0 of the compression rate consists of first linearly interpolating between two adjacent data points so that the sum of original data points (screen width * compression rate) and interpolated data points is equal to the screen size. For greater than 1.0 of the compression rate, the Signal Editor will detect maxima (minimum- and maximum-valued

data) in an interval of the compression rate and vertically draw the maxima in the current point by connecting either the minimum or maximum data point of the neighboring interval of the compression rate. This display method is used in all windows of the Signal Editor.

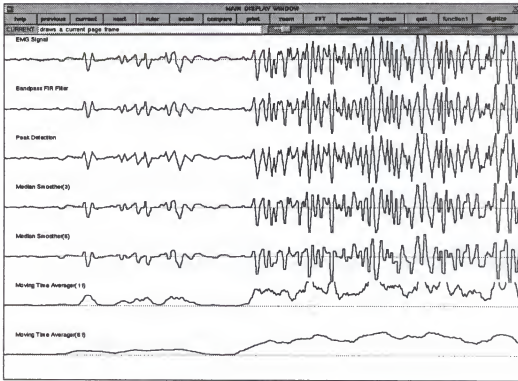


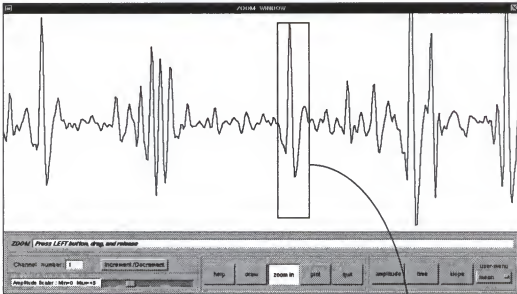
Figure 4.7. Main display window

In this window, the user can move to the next and previous frame segment of the data, or repeat the display of the current frame segment, change the signal amplitude level, draw rulers manually or automatically, print out the window, and randomly jump to a selected file position by moving the slider of the file pointer to a certain location. By clicking the “option” button, the user can change the number of the ruler, signal amplitude, and the channel title in the Parameter Setup window which is shown in Figure 4.15 (c). The title of channels should be in the header file of the associated data. The digitizing data can also be displayed by clicking the “digitize” button when the analog-digital board is installed in the NeXTcube. All subwindows of zoom, FFT, acquisition, and comparison windows are activated from this window.

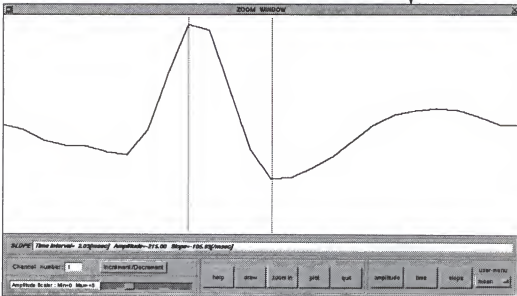
Zoom window. The most frequently issued subwindow for signal measurements is the Zoom window shown in Figure 4.8, in which the user can closely scrutinize small portions of the signal. The Zoom window works as a software microscope by enlarging the time scale and changing amplitude scales without modifying their original data value. The system automatically displays the measurements, i.e., time, duration, slope, etc., in the message box of the window. The points used for measurements are clearly line-marked on the screen. In a low compression rate, less than 1.0, the user can measure parameters with/without zooming. For a large compression rate, greater than 2.0, the user can also measure parameters without zooming but the system sends a warning message to reduce the compression rate for correct analysis. In this case, the user first selects an appropriate segment so that the length of the selected one is less than the screen width, and zooms in to measure accurate parameters. For the zoomed segment not smaller than the screen width, the measurement cannot be performed. Therefore, the system suggests reducing the length of the zoom segment. The user should choose a channel from the display channels in the Main Display window before measuring the signal.

This window is considerably valuable for precisely measuring signal time, amplitude, period, and slope, which are general measurements for any type of signal analysis. This window also has a special pop-up button for the user-dedicated measurement. It allows the user to measure his special parameters such as calculation of the mean value which is currently coded in the system. The user can change or add programming codes for specific measures in an appropriate area of "userMeasure.m" and/or "userMeasure.h" files which are defined in the Signal Editor. The header and implementation files are shown in Figure 4.9. The "data" in the figure is a pointer to input data to be measured, and the "length" means the size of the input data. The "buttonTitle" is a pointer of the title name to be written in the associated pop-up button and the "buf", containing the result of user-measurement, is a string pointer to be displayed in the message box. There are five available options for user-oriented measurements. After writing codes

and compiling, the user can access his own measure routines by clicking associated pop-up buttons under the “user-menu” title in the window.



(a) Window before zooming



(b) Window after zooming

Figure 4.8. Zoom window

```

#import <objc/Object.h>
@interface UserDefinedMeasure:Object
{
}
+ new;
- userMeasure1:(int *)data :(int)length :(char *)buttonTitle :(char *)buf;
- userMeasure2:(int *)data :(int)length :(char *)buttonTitle :(char *)buf;

- userMeasure5:(int *)data :(int)length :(char *)buttonTitle :(char *)buf;
@end

```

(a) userMeasure.h file

```

#import "userMeasure.h"
@implementation UserDefinedMeasure
+ new
{
    self = [super new];
    return self;
}
- userMeasure1:(int *)data :(int)length :(char *)buttonTitle :(char *)buf;
{
    // write program code 1 for measuring user-specific parameters
    return self;
}
- userMeasure2:(int *)data :(int)length :(char *)buttonTitle :(char *)buf;
{
    // write program code 2 for measuring user-specific parameters
    return self;
}

- userMeasure5:(int *)data :(int)length :(char *)buttonTitle :(char *)buf;
{
    // write program code 5 for measuring user-specific parameters
    return self;
}
@end

```

(b) userMeasure.m file

Figure 4.9. The “userMeasure” files for measuring user-defined parameters

FFT window. A power spectrum representation is sometimes useful in estimating the dominant frequency components of the relatively stationary segment or long-term events, or in evaluating noise-reduction algorithms. In the power spectrum window shown in Figure 4.10, the user selects the segment of interest, and the Signal Editor calculates and displays the estimated power spectrum.

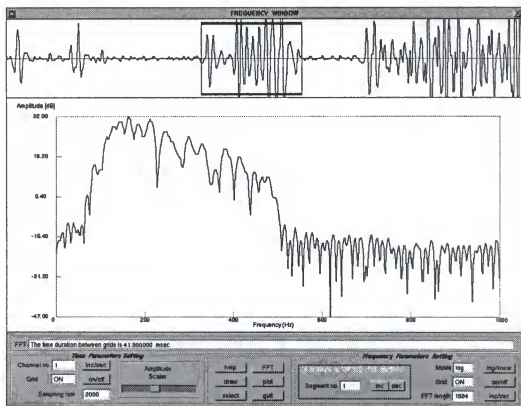


Figure 4.10. FFT window

The power spectrum estimation is computed by averaging modified periodograms, i.e., the Welch Method [Op75]. A data sequence, $x(n)$, $0 \leq n \leq (N-1)$, is divided into K segments of M samples each so that $N = K \times M$. The rectangular window, $w(n)$, is applied directly to the data samples before computation of the periodogram. The K periodograms are

$$J_M^i(w) = \frac{1}{C} \cdot \left| \sum_{n=0}^{M-1} x^i(n) \cdot w(n) \cdot e^{-jwn} \right|^2$$

where, $i = 1, 2, \dots, K$, C is constant. The spectrum estimate of the Welch method is defined as

$$P(w) = \frac{1}{K} \sum_{i=1}^K J_M^i(w)$$

The amplitude scale which is represented by either logarithm or linear mode is automatically determined by detecting the maximum and minimum of $P(w)$. The

horizontal axis displays frequencies of zero to $f_s/2$ (f_s =sampling frequency) based on the sampling rate. The number, M , of overlapping segments is automatically determined unless the user does not select appropriately. When the selected data segment is not long enough to implement the Welch procedure, zero-padding is performed by adding zeros (FFT length - data segment length) to the end of the data segment, so the number of overlapping segment reduces to one. The FFT window lengths that the user can choose are 64, 128, 256, 512, and 1024.

Acquisition window. Long-term signal monitoring produces large amounts of data. In some applications such as the diagnosis of epilepsy in a biomedical signal, the user is interested in the portions of data having meaningful signal patterns, which are scattered throughout the entire data file. The user sometimes needs these kinds of special waveforms to test his/her algorithm. Thus, the user can select relevant segments in the Acquisition window shown in Figure 4.11.

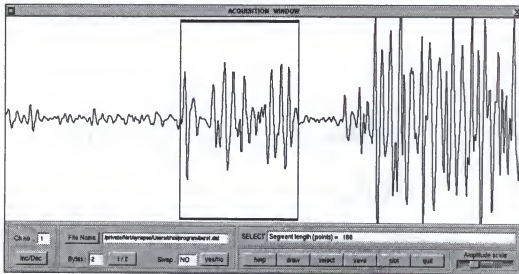


Figure 4.11. Acquisition window

The appropriate segments are extracted and stored sequentially in a new data file, or appended on the existing file, by selecting the file in the File Save window shown in

Figure 4.15(b). The data formats, i.e., number of data bytes and swap status, to be saved is the same as those of the original signal unless the user changes. The user can redisplay them in the Main Display window without exiting the Signal Editor. This capability also simplifies the validation of detection algorithms by providing specific test patterns.

Comparison window. The window shown in Figure 4.12 presents the differences between the two display channels of the Main Display window. This is especially useful for visually evaluating a noise reduction algorithm or for estimating how much a filter routine suppresses unwanted signal components in a time domain. It can also serve to compare outputs of different algorithms qualitatively. In this example, an original signal and the signal after lowpass filtering overlap with different gray levels in the upper side of the display area, and the difference between them is shown in the lower side. The user can select any combination of the two display channels to be compared.

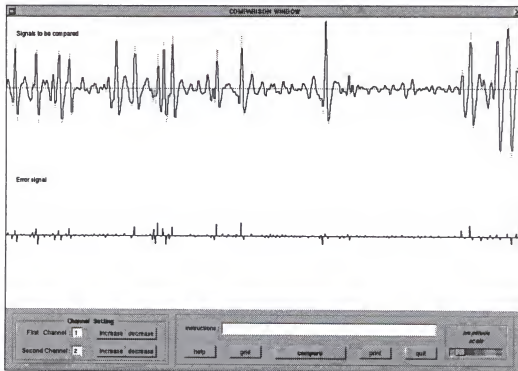


Figure 4.12. Comparison window

Function update window. In this window, the user can access the Function Library by either adding the user-developed algorithms or deleting the built-in functions.

Much of object-oriented programming consists of defining new object classes. In objective-C in the NeXT computer, classes are defined in two files: (1) A file that declares the interface to the new class. (2) A file that actually defines the class (contains the code that implements it). Each class requires both files; the declaration of the interface and the class implementation cannot be in the same file. For each class, it is customary to have separate interface and implementation files. The implementation file has a “.m” suffix, indicating that it contains an Objective-C source code. The interface file can be assigned with any other extension. Because it is included in other source files, the interface file usually has the “.h” suffix which is the typical name of header files. Therefore, when the user adds functions, the AFES generates associated function objects and connects those objects to the system. New functions are automatically registered in the Function Library and the AFES makes both the implementation file and the interface file which are dedicated to the new function. Thus, the user can implement his/her algorithm by writing a code in the implementation file which conforms to the system-defined input/output format with fixed data length.

<pre>#import <objc/Object.h> @interface FIR:Object { } + new; - (void)process:(int *)input :(int *)output :(int)data_length :(float *)parameter; @end</pre>	<pre>#import "FIR.h" @implementation FIR:Object + new { self = [super new]; return self; } - (void)process:(int *)input :(int *)output :(int)data_length :(float *)parameter { // write your code in this area } @end</pre>
(a) Interface file	(b) Implementation file

Figure 4.13. Example of function files of FIR object

An example of files is shown in Figure 4.13. The first line of the declaration in the interface file presents the new class name (FIR) and links it to its super class (Object). The superclass defines the position of the new class in the inheritance hierarchy. Following the class declaration, methods for the class are declared. The names of methods that can be used by class objects are preceded by "+ new." The methods in which instances of a class can be used are called instance methods, "- (void)process:::" The purpose of the interface file is to declare the new class to other source modules. It contains all the information they need to know about the class. A class definition is structured very much like its declaration. Every implementation file imports its own interface (e.g., #import "FIR.h"). The definition of the instance method has all the instance variables of a potential receiving object within its scope. It can refer to them simply by name (e.g, input, output, etc.). Methods can be added to a class by declaring them in the interface file and defining them in the implementation file so that the user can increase his own methods to have a structured programming paradigm. The Function Update window is shown in Figure 4.14.

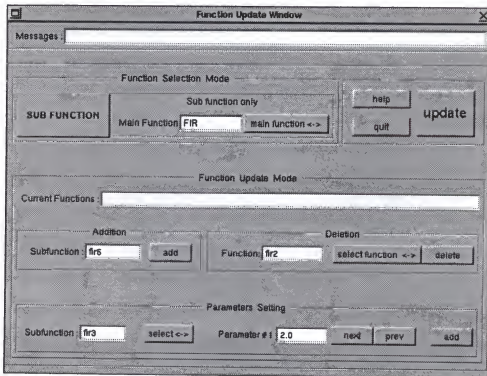


Figure 4.14. Function update window

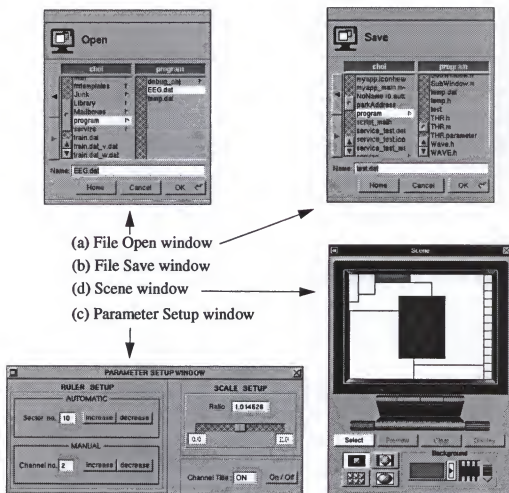


Figure 4.15. Other windows in Signal Editor

4.4 Implementation of the Segmentation System

The segmentation model which consists of signal segmentation and boundary estimation was discussed. This section describes the implementation of the automated segmentation system for the respiratory EMG signals. Two major parts are discussed: the functional description of the system and signal visualization based on the display windows. The system was developed mainly using the IB and Application Kit. After designing all interface objects in the IB, the functional programming codes are written in the corresponding methods of implementation files.

collection and neural network learning). Every window has its own window class and it is activated by selecting an associated box of the top level menu which is shown in Figure 4.18(a). Since the functional module is a method in the object-oriented structure, it can be accessed by sending a message which consists of the name of the method and any arguments that are passed to it.

Both an EMG data file which consists of two channels of signals (PCA and DEMG) and the Flow signal file are read into the Input Buffer. The first half of the data file is PCA signal and the second half is DEMG signal of binary format. The header file which contains the information of the data file and some parameters to be used in signal processing is also read. An example of the header file is shown in Figure 4.17. Data of Input Buffer are passed through noise rejection, signal segmentation, and boundaries detection, resulting in boundary candidates. Then they are estimated by a discriminant function of the neural network, which is determined in the boundary classification module before new data are analyzed.

```

/* This file contains all information of the data file */
/* Don't change the order and position of each item */
Number of channels : 2
Sampling rate      : 2000
Number of data bytes : 2
Swap bytes         : 1
Highpass filtering : b          /* p for PCA, d for DEMG, b for BOTH, n for NO */
Noise canceler     : d 0.011 /* p for PCA, d for DEMG, b for BOTH, n for NO */
Thld Coef. of PCA   : 4.0 1.5 1.5 1.5 /* 1st two for cess, rest for onset. */
Thld Coef. of DEMG : 4.0 1.5 1.5 1.5 /* 1st two for cess, rest for onset. */

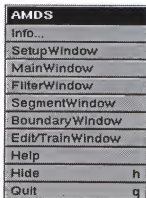
```

Figure 4.17. Example of header file

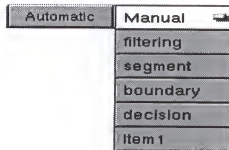
The user can control any commands in the windows such as the selection of a data file, operational mode, and display configuration. The communication between functional modules is through files, that is, the data and information files. Since each functional module has its input and output buffer, the file is read into the buffer and the algorithm

result is saved into another file for the next functional module. The window reads result files of the functional modules and displays various forms by the user command.

The system was designed with two operational modes, automatic and functional. In the automatic mode, four functional modules (noise rejection, signal segmentation, boundaries detection, and boundary decision), which are directly associated with the proposed segmentation model are performed at one time. In the functional mode, the system can run all functional modules one-by-one. With either way, the intermediate results are displayed on several windows. The operational mode can be chosen by pressing "automatic" or "manual" button in the Main window which is shown in Figure 4.18(b). The relationship of functional modules and display windows are as follows: the Filter window for the noise rejection module, the Segmentation window for the signal segmentation module, the Boundary window for the boundaries detection module, the Main window for the boundary decision module, and the Event Edit window for the boundary editing and boundary classification module.



(a) Top-level menus for window selection



(b) Operational mode selection

Figure 4.18. Menus for the selection of display windows and operational mode

Following are descriptions of the functional models and the windows.

4.4.2 Functional Description

The automated segmentation system consists of six functional modules which communicate through files. The characteristic of each functional module is discussed. The relationship between files and functional modules, and between files and display windows is described.

Noise rejection module. The purpose of this functional module is to reject noise. There are three messages of object oriented structure. In the first message, a linear phase FIR bandpass filter is implemented. A digital filter is a discrete-time linear shift invariant system. The input and output of digital filters are related by the convolution sum of an equation like

$$y(n) = \sum_{k=1}^N h(k) x(n-k) = h(n) * x(n)$$

where $h(n)$ is an impulse response, $x(n)$ is input, $y(n)$ is filtered output, and N is the length of the impulse response. In this research, the 63th order impulse response was used. Since the time delay in output is associated with the length of impulse response used, the alignment is done by shifting $(N-1)/2$ points in computation. The input and output file used in this message is “-.dar” and “-.bp.”

The second message deals with a notch filter, i.e., an adaptive noise canceler(ANC). The adaptation time in the ANC algorithm has been shown to be dependent on the change rate of the signal energy as well as the adaptation constant, μ . It also shows that in the signal change from large energy to small energy or vice versa, the output of the ANC filter in the time domain is not the same based on the direction of signal processing because the effect of adaptation on the signal energy change is different. Since forward data processing produces clearer signal changes than reverse data processing for the transition from the background to the burst activity, it can be used for detection of the onset boundary point. For the signal changes from the burst activity to the background (or for the detection of cessation boundary point), backward data processing is better for detecting the boundary.

Therefore, we decided to apply the ANC algorithm in both directions to get the clearer boundary signal. Results of forward and reverse data processing are saved into files of “*-.ancf*” and “*-.ancr*.”

In the third message, two files are combined into a file named “*-.ancl*.” The signal combination is based on the zero-crossings of Flow signal, which was explained in chapter 2. Thus, all zero-crossings are detected and saved into “*-.pnr*” file. The general idea is that the period of a breath is divided into two parts. One part is a segment of $(nF1+F3) / 2$ and $(F2 + (F3-F2) / 4)$ in Figure 2.19. The segment is taken from “*-.ancr*” file, in which the cessational point is included. The other is a segment of $(F2 + (F3-F2) / 4)$ of current breath and $(nF1+F3) / 2$ of the next breath. The segment is taken from “*-.ancf*” file, in which the onset point is included. The above two segments are put together and saved into the “*-.ancl*” file. This procedure is repeated for other breaths. Figure 4.19 depicts signal combining graphically.

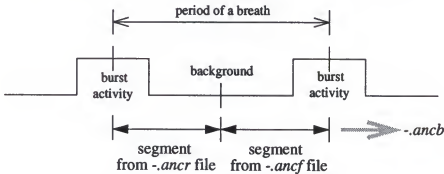


Figure 4.19. Description of signal combining

The filtering output of both directions is shown in the Filter window. The above signal combining is done independently for PCA and DEMG. If the ANC filtering is not needed to apply, the bandpass filtered data is saved directly into the “*-.ancl*.” The information, whether the ANC is applied or not, is in the header file.

Signal segmentation module. The signals are divided into three segments: background, transitional, and burst activity. This module consists of three function calls. The first message handles the computation of the Hanning-windowed moving time

average. It reads the “-.*ancl*” data file and computes MTAs of different window lengths such as 21 and 81. The MTAs are saved into “-.*mta21*” and “-.*mta81*” files which consist of PCA and DEMG in sequence. Since the amplitude of the MTA is large, it is divided by the MTA window length. The second message takes the derivative of MTA81 and saves it in the “-.*der*” file. For the derivative, the central finite difference with step size of 5 points was used.

The third message detects the threshold of onset and cessation for the recognition of burst activity based on the procedures explained in chapter 2. The threshold was defined as $k \times$ average of the minimum background. Since the background level and the energy ratio between the burst activity and the background are different among subjects, the k value is hard to fix for all subjects. Thus, the value of k is specified in four different ways in the header file (refer to Figure 4.17) so that it is given separately for the onset and cessation of PCA and DEMG to come up with the variability of the background signal. After detecting thresholds, the program determines the transitional segments. First, the coarse cessation segments are determined followed by the coarse onset segments. Within the coarse transitional segments, the fine transitional segments for cessation are determined followed by the fine transitional segments for onset. The graphical description of thresholds and transitional segments is shown in Figure 4.20. The result of signal segmentation is shown in the Segmentation window.

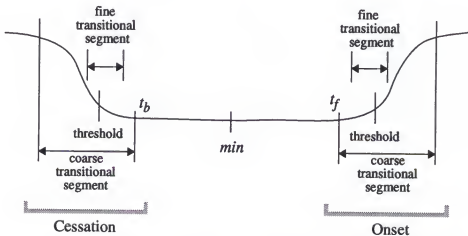


Figure 4.20. Graphical presentation of thresholds and transitional segments

In detecting the thresholds and the transitional segments, the zero-crossings of the Flow signal are required. Thus, the “-.pnf” file is read when the third message is instantiated. After determining the transitional segments, all minimum points of the background, thresholds in time, and transitional points of onset and cessation are appended to the “-.pnf” file which is shown in Figure 4.21. The symbol, *A*, in the figure means the total number of breaths, *B* indicates zero-crossings of the Flow signal, *C* and *D* indicate threshold points of PCA and DEMG, *E* and *F* indicate minimum points in the background of PCA and DEMG, *G* and *H* indicate points of the coarse transitional segments of PCA and DEMG, and *I* and *J* indicate points of the fine transitional segments of PCA and DEMG.

37		<i>A</i>
0 365 1000 1445 1975 2605 3080 3610 4250 4786 5306 5946 6424 6974 7634 8181 8731 9381 54663 55193 55843 56400 56930 57570 58011 58561 59181 59612 60192 60852 61309 61829 62469		<i>B</i>
822 1427 2420 3092 4060 4803 5761 6421 7454 8218 9186 10130 11138 11909 12892 13606 47262 47998 48936 49652 50590 52271 52952 53905 54730 55648 56447 57370 58073 58981 59694		<i>C</i>
797 1775 2435 3425 4075 5096 5766 6774 7449 8521 9181 10443 11138 12212 12877 13914 47272 48291 48951 49960 50595 52266 53270 53895 55008 55658 56750 57380 58366 59006 59997		<i>D</i>
1332 3040 4680 6376 8154 10006 11833 13497 15199 17002 18765 46293 47892 49561 51225 52891 54625 56368 58000 59621		<i>E</i>
1232 2940 4680 6476 7554 9306 11333 12997 14999 16702 18265 45693 47392 49161 50825 52391 54025 56068 57500 59921		<i>F</i>
677 696 833 1383 1526 1541 2347 2388 2456 3069 3102 3161 4005 4026 4093 4774 4842 4873 56433 56514 56561 57313 57338 57399 58063 58154 58176 58888 58923 59017 59674 59732 59759		<i>G</i>
750 774 823 1759 1803 1827 2389 2417 2455 3409 3446 3464 3987 4047 4098 5094 5124 5144 56716 56767 56792 57333 57363 57402 58350 58386 58407 58959 58982 59025 59976 60018 60038		<i>H</i>
777 831 1402 1457 2393 2452 3081 3103 4053 4072 4787 4827 5713 5782 6428 6455 7429 53929 54723 54752 55616 55659 56441 56474 57341 57398 58078 58101 58964 59016 59685 59718		<i>I</i>
775 826 1768 1799 2406 2443 3425 3452 4037 4097 5105 5138 5743 5769 6760 6808 7425 7473 53924 55003 55046 55645 55692 56722 56779 57362 57398 58370 58401 58969 59008 59981 60030		<i>J</i>

Figure 4.21. Example of data points in “-.pnf” file

Boundaries detection module. Boundary candidates in onset and cessation transitional segments are detected. After five energy functions, combinations of MTAs and

STEs are calculated and they need to be normalized because the background level between breaths varies. As shown in Figure 4.22, five equal segments of 30 msec centered at an endpoint of the coarse transitional segment are selected. The information of the transitional segment are read from “*-.pnt.*” The average amplitude of all segments is computed and the minimum-valued segment is chosen as a representative level of the onset or cessation background. Based on the level of the minimum-valued segment, the signal in the transitional segment is quantized into two values. If the signal amplitude is less than five times the minimum-valued segment, the signal is set to zero. Otherwise, it is set to a constant value such as 200. An example of the energy contour is shown in Figure 4.22. Since all boundaries have been shown to be in the interval of transitional segments, especially between f_1 and t_2 , the boundary candidates are detected in the interval. The signals out of the interval are set to zero for display.

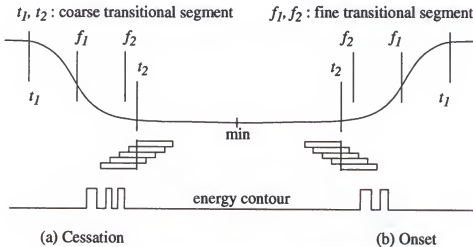


Figure 4.22. Determination of boundary candidates

Whenever the signal is changed from a higher value to zero or vice versa in energy contours, the amplitude changing point becomes a boundary candidate. Usually there are several boundary candidates in a transitional segment. The same timing points of boundary candidates may be detected from different energy functions. In this case, the same boundary candidates are merged into one. The boundary candidates are collected by four

groups such as onset and cessation of PCA and DEMG. All boundary candidates are saved into the “-.est” file with the following information: PCA or DEMG, onset or cessation, minimum point in the background, t_2 and f_2 of coarse and fine transitional segments, number of boundary candidates, and their points for use in the boundary decision functional module. Five energy functions are saved into five different “-.ste” files. Boundary candidates of five energy functions are shown in the Boundary window.

Boundary Decision module. Boundary candidates detected from the boundary detection module are estimated through the neural network. The boundary candidates are read from the “-.est” file. Before being sent to the network, the boundary candidates are quantized into several discrete levels by comparing the background amplitude. Two background segments were decided to be used for the stable background level. For every boundary candidate, the minimum (*min* in Figure 4.22) in the background and a point (t_2 in Figure 4.22) of the coarse transitional segment which is nearest to the boundary candidate of interest are found.

All peaks are detected in the interval of $min-30$ msec and $min+30$ msec and then all modified line lengths, i.e., duration-slopes, between adjacent peaks, are computed based on the rules which were defined in section 3.3. After the duration-slopes are sorted by magnitude, the average background level is obtained by rejecting the two highest and the two lowest line segments to be a representative value for the background. The above procedures are repeated for t_2 of the coarse transitional segment, that is, an interval of 30 msec to the right from t_2 for the cessation boundary candidate or an interval of 30 msec to the left from t_2 for the onset candidate. This gives another background level for the same boundary candidate. Since it has been shown that these two background levels are not always the same due to the unstable background phenomena, the *final background level* is determined by the combination of the two background levels. The general rule was explained in section 3.3. The final background level is not the same for onset and cessation boundary candidate because the background level of t_2 is different although the same

background level of *min* is used.

For selected boundary candidates, duration-slopes of peak-to-peak of 100 msec intervals centered at the boundary candidates are computed. Based on the rules in section 3.3, the duration-slopes in the interval are compared with the final background level and they are quantized into 7 different (and discrete) levels. Twenty quantized values around the boundary candidates of interest, that is, 10 before and 10 after a boundary candidate, are chosen and saved into an “*-.ann*” file.

When the message of the boundary decision function is called, the neural network is first initialized by reading the network architecture and learning parameters from “*train.dat_v.dat*” and network weights and thresholds from the “*train.dat_w.dat*” file. The quantized values are read from the “*-.ann*” file and used for the input pattern to the neural network. Thus, the boundary candidates are estimated by the boundary discriminant function which is determined by network weights. The estimated boundaries are saved into a “*-.rlt*” file. The boundary estimates with probability can be displayed in the Main window.

Boundary editing module. Two EMGers are requested to mark their boundary points on the EMG paper tracing which contains two channels of signal. The boundaries are determined based on the EMGer’s visual discrimination. The human-detected boundaries on the paper are needed to quantify in time to compare with system-detected boundaries. The quantification is performed in the Event Edit window. The boundaries are collected with four different groups such as both the onset and cessation of PCA and DEMG signals. Thus, the system performance can be evaluated by both signal and transitional types.

All quantified boundary points are saved into an “*-.event*” file with the following information: PCA or DEMG, onset or cessation, and a quantized boundary point. The boundaries can be displayed in the Event Edit window to make sure that the edited boundaries are correct. Wrong boundaries can be deleted and new boundaries can be added.

Boundary classification module. (1) Collection of training data: The representative training data sets are required for the network to derive the boundary discriminant function for unbiased estimation. The training sets can be collected in the Event Edit window. Clear transitional segments are selected based on a visual discrimination by browsing EMG signals and then boundary points and non-boundary points from the onset or cessation of PCA and DEMG are chosen. The training data sets are chosen by Dr. A. A. Hutchison, Dr. J. A. Wozniak, and H. G. Choi.

When either a boundary or a non-boundary point is selected, the whole procedure of getting input patterns for the neural network is the same as that of the boundary decision module. For selected points, their duration-slopes of 100 msec intervals centered at the points are computed in the same way. The duration-slopes in the interval are compared with the final background level and they are quantized into 7 different (and discrete) levels. Twenty discrete values around a point of interest are chosen and saved into a "*train.dat*" file with class information, that is, boundary class or non-boundary class. These points are used to train the neural network in the network learning module.

(2) Network learning: The neural network derives the general boundary discriminant by adapting the network parameters through training data sets. When the learning routine starts after training data sets are read from the "*train.dat*" file, we set the momentum of 0.9 and the learning rate of 0.7, the maximum system error of 0.00005, the maximum individual error of 0.000005, the number of iterations of 2000, and the number of hidden layers of 1 and their processing units of 18.

In network initialization, the dynamic storage for the network is allocated and weights with random numbers between -0.5 and 0.5 are assigned, and then network error using backpropagation algorithm is computed. The threshold of the processing unit is treated as a weight of link from a virtual node whose value is unity. If the system error fails decrease the maximum system error until the maximum number of iterations is reached, the learning is terminated. For successful learning, three files are created, a "*train.dat_w.dat*"

for saving weights and thresholds learned from training data sets, a “*train.dat_v.dat*” for network architecture and learning parameters, and a “*criter.dat*” for saving the changes of the system error to show how fast the system converges to the characteristics of the training data sets. This network learning is performed in the Event Edit window. The network training is needed only one time. Once the network training is successfully finished and whenever the system starts to run, all necessary network parameters can be read from the above files without learning again. Thus, the system-detected boundaries or human-detected boundaries can be rated through the network.

Files in display windows and functional modules. All communication between functional modules, and between functional modules and display windows are performed through the files. Associated files of windows and functional modules are shown in Figure 4.23 and Figure 4.24.

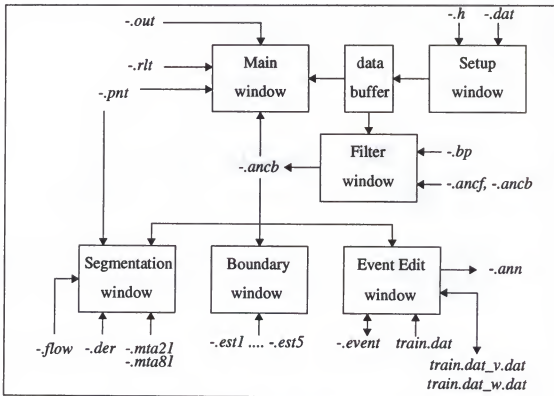


Figure 4.23. Files which are related to display windows

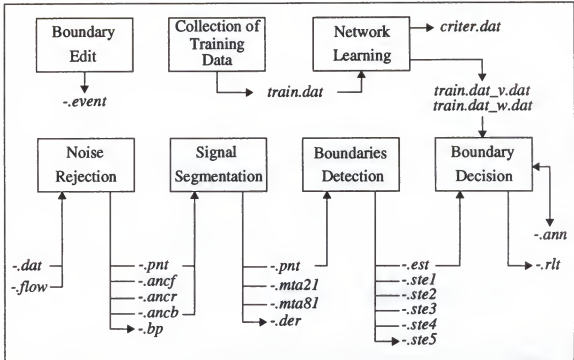


Figure 4.24. Files which are related to functional modules

4.4.3 Signal Visualization

There are five display windows; the Main window for displaying boundary estimates and transitional segments, the Filter window for displaying filtered signals, the Segmentation window for the result of signal segmentation, the Boundary window to display boundary candidates detected in the transitional segment, and the Event Edit window for editing of human-detected boundaries, collecting and training data. Signal visualization on several windows is quite useful to evaluate the intermediate results and to facilitate the system development by helping to determine several parameters used in the system. The windows are independently operated in terms of data handling, scrolling, display configuration, etc. This section explains the purpose of windows and their features.

Setup window. Before running other windows, the segmentation system should be initialized by specifying a data file name and a compression rate. When the file is accessed

through the File Open window, the system seeks its header file which contains information on the data file such as the number of data channel, sampling rate, number of data bytes, etc. If either the data file or header file fails to access, the system cannot be initialized, resulting in errors of functional manipulation in other windows.

The compression rate is related to the display size on a screen. The number of data points on a display channel is set to 1000 pixels for all windows. For the compression rate of n , the data segment to be displayed on the screen is $1000*n$ points which corresponds to $(1000*n / \text{sampling frequency})$ in time. The minimum and maximum compression rate is 0.1 and 10.0 with a step size of 0.1. For the sampling frequency of 2000 Hz, the time range spans from $100*0.5$ msec at a minimum compression rate to $10000*0.5$ msec at a maximum compression rate. The Setup window is shown in Figure 4.25.

Main window. The segmentation algorithm can be run by either the automatic or functional mode by clicking the associated button named "automatic" or "manual." Four signals are drawn in the following order: an original PCA signal, its filtered signal, an original DEMG signal, and its filtered signal. The system results are also shown, such as boundary estimates with probabilities, points of transitional segments, and the final boundary. The Main window is shown in Figure 4.26. The interval of $C1$ and $C2$ in the figure denote the coarse transitional segment while $B1$ and $B2$ denote the fine transitional segment. The boundary estimates with probabilities are presented on the display channel of filtered signals when the segmentation system is finished successfully. The point, F , in this figure presents the final boundary point for the given transitional segment. The data segment corresponding to the compression rate can be displayed sequentially by clicking the "next" or "prev" button, or randomly by moving a file pointer in the slider.

Filter window. The Filter window is shown in Figure 4.27. Original signals and filtered signals are displayed in several ways. By combining the display mode, "BP/ANC" and "pca/demg", the channel signals which would be displayed are shown in Table 4.1.

Table 4.1. Signal channels to be displayed in Filter window

BP/ANC	pca/demg	display channels
BP	pca demg both	PCA, PCA bp DEMG, DEMG bp PCA, PCA bp, DEMG, DEMG bp
ANC	pca demg both	PCA, PCA anc DEMG, DEMG anc PCA, PCA anc, DEMG, DEMG anc
BOTH	pca demg both	PCA, PCA bp, DEMG bp DEMG, DEMG bp, DEMG anc PCA, PCA bp, PCA anc, DEMG, DEMG bp, DEMG anc

where, bp means the bandpass filtered signal and
anc means output through the adaptive noise canceler

In particular, when the “forward” button is clicked, two signals through the notch filter are displayed; one processed with a forward direction of data and another processed with a reverse direction when an adaptive noise canceler is applied.

Segmentation window. The Segmentation window is shown in Figure 4.28. It displays the original EMG signals, the MTA signals of window length 21 and 81, and the derivative signal. In this window, the signal amplitude of all channels’ data is checked. Thus, the thresholds used to recognize the burst activity can be estimated based on the level of the background and the burst activity on MTA signals, for example, in the determination of the parameter, k_i in $\text{mean} + k_i \cdot \text{SD}$. Segmentation results are shown in the window, i.e., the coarse transitional segment (*C1* and *C2* in figure), the fine transitional segment (*B1* and *B2*), and the threshold (*T*) of the burst activity. In addition, the Flow signal can also be drawn with zero-crossings by selecting the “optional” pop-up menu.

Boundary window. Figure 4.29 shows the Boundary window. The segmentation system implements five different energy functions, a combination of MTA and STE functions using various window size, to detect all probable boundary candidates. The

number and timing points of boundary candidates are different between energy functions, sometimes occurring at the same timing point. The window shows the boundary candidates detected in each energy function. The signal transitions, i.e., low to high or high to low, in the energy contours are the boundary candidates. The maximum channel number of the energy function is five.

Event edit window. Two distinct functions are performed in this window, event edit and network training. The human-detected boundaries are required to quantize in time to be compared with system-detected boundaries. In the event edit functional module, the boundary event is divided into four different categories: PCA onset, PCA cessation, DEMG onset, and DEMG cessation. Thus, the user can input his boundary by selecting a specific category. The window displays boundary points which are already edited. Thus, wrong boundary input can be corrected by deleting an old one and editing a new one.

In the network training functional module, the training data sets are collected in a similar way to boundary event editing except that instead of categorizing, the boundary or non-boundary class information is added and then it is used for the network training. In the learning session, maximum system error and individual error, and maximum iteration numbers should be specified. Thus, parameters of the neural network as boundary classification are used to rate the boundary candidates. In this window, the manually selected boundary points can also be estimated. The Event Edit window is shown in Figure 4.30.

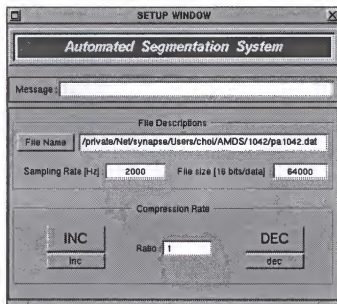


Figure 4.25. Setup window

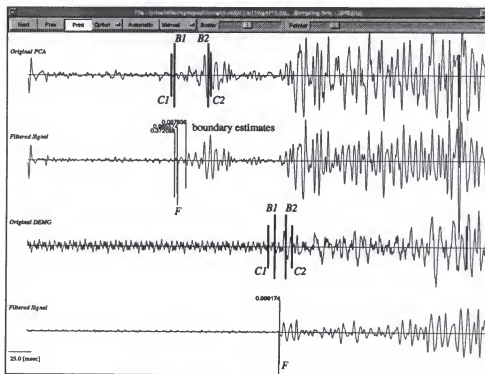


Figure 4.26. Main window

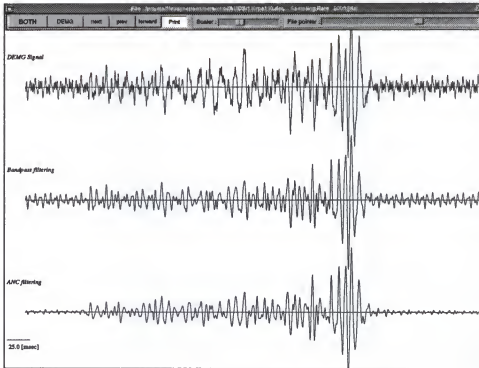


Figure 4.27. Filter window

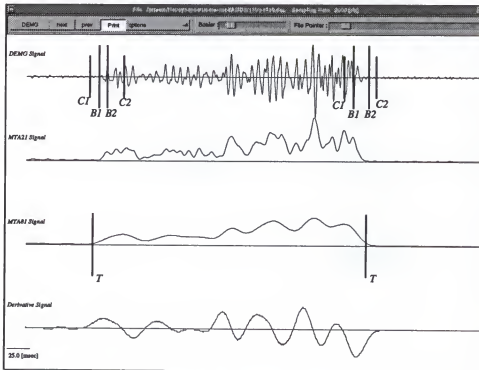


Figure 4.28. Segmentation window

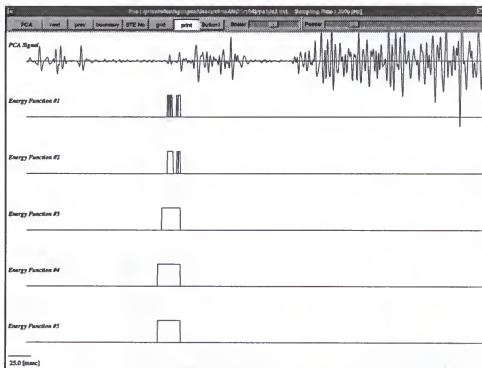


Figure 4.29. Boundary window

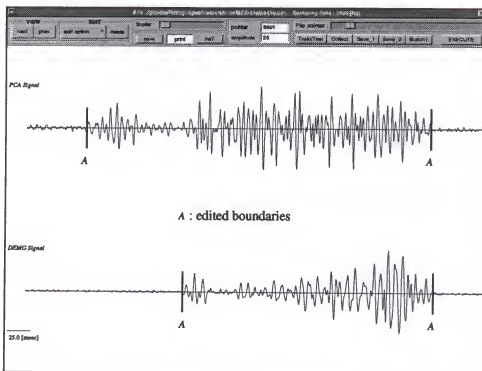


Figure 4.30. Event edit window

4.5 Discussions

In biomedical applications, computer systems can be used for the visualization of multichannel data. They can replace the currently used paper-polygraph systems. However, computer-based systems are much more powerful and flexible in signal manipulation. The user can examine and modify the data, quantify it, and enhance some of its features through signal processing. The computer provides the means of applying a broad class of signal processing algorithms and of visualizing the effects of the algorithms on the data. It also provides the means of quantifying pre-defined parameters, random access to any data segment, data archives, and printouts. These characteristics are the basis for the much more productive interaction with the data when the computer is utilized.

Functionality in the computer-based system is easily achieved through an object-oriented programming paradigm. For example, according to clinical importance, a software architecture can be designed to develop automated waveform detectors of clinically important signal components. For the design of waveform detectors, the Signal Editor can provide a prototype for algorithm implementation and for validating new detectors quantitatively and qualitatively. In this sense, the Signal Editor is quite useful in biological signal analysis. Generally speaking, an obvious role of the computer-based system like the Signal Editor is the creation of environments where the expert can visualize the data by mimicking the paper output of the polygraph, accurately measure the parameters by manipulating a large amount of data (i.e., storage and retrieval of biosignal), and develop an automated detection methods that can detect important events.

All algorithms implemented in the automated segmentation system of respiratory EMG signals were tested in the Signal Editor. After the algorithms were evaluated or validated, the appropriate programming codes were copied into the segmentation system. The Signal Editor is an object-oriented implementation for the NeXT computer of T DAT [Pa90]. This program is being utilized by the Neuroengineering laboratory at the University of Florida.

The object programming style in both systems makes system management easy and flexible in terms of the addition of new functions, function update, and maintenance of the system.

CHAPTER 5

SYSTEM EVALUATION AND RESULTS

This chapter describes data collection and system evaluation. The developed system was tested using five sets of EMG data record digitized from different subjects. Boundaries detected by the system are compared with those detected by two EMGers to assess the system performance in terms of accuracy of boundary detection. The transitional segments are also evaluated.

5.1 Data Acquisition and Transfer

Florida Cross lambs were used for the study. Their birthweight (mean \pm standard deviation) was 3.7 ± 0.7 kilograms. Their gestational age at birth was 140 ± 0.5 days and they were studied at 7 ± 0.5 days postoperatively. Using sterile procedures and under halothane anesthesia, lambs were operated on at a gestational age of 132 or 133 days. Stainless steel wire electrodes (Cooner wire) were implanted into the respiratory muscles. The airflow signal was measured via a Rudolph pneumotachograph, a 2 cm H₂O Validyne differential pressure transducer and a Gould transducer amplifier. Volume was obtained by electronic integration of the flow signal. Differential transducers used for measuring the airflow signal had 3 dB/octave amplitude loss in excess of 10Hz. Respiratory muscle EMGs were measured using Gould pre-amplifiers and universal amplifiers, with appropriate specification [Br84]. Cutoff frequencies of the amplifier were 30 and 1000Hz and an analog notch filter was employed to reject 60Hz noise.

When spontaneous respiration was established, recordings were made of the changing patterns of respiration until a normal sinusoidal pattern was established. Expiratory flow characteristics were used to define the patterns of respiration. At the onset

of respiration at birth, the premature lambs are frequently asphyxiated, and the pattern of respiration was irregular. After the period of irregular breathing, a regular breathing pattern (called "Pattern A") commenced. This was a sustained and spontaneous respiration characterized by a period of severe retardation of expiratory flow (refer to Figure 1.2). The proportion of retarded expiratory time was decreased with advancing postnatal age. After a period of transition, the expiratory airflow pattern was more sinusoidal in nature (called "Pattern B"), showing little retardation of flow. During transition, the number of breaths with severe expiratory flow retardation decreased and the number with a more sinusoidal pattern increased.

The experimental respiratory EMG data was recorded on a magnetic tape recorder and simultaneously plotted on a polygraph in Dr. Hutchison's laboratory at the University of Florida (UF). Since Pattern A data segments are used in this research, representative data segments which contain the characteristic of breath patterns are found by examining the polygraph and by tracing the replayed magnetic tape on an oscilloscope. The selected segments are digitized onto a hard disk by a personal computer using the STREAMER software and the Metrobyte DAS-20 ADC (analog-to-digital converter) with 8 differential channels, resolution of 12 bits, input range of ± 5 volts, and DMA (Direct Memory Access) data transfer mode. Considering that the maximum frequency of EMG signals is about 500 Hz, the sampling frequency was set to 2000 Hz. The total number of digitized channels is 8, which mainly consist of voice, airflow, volume, pressure (upper airway and mouth), EMG signals (PCA, DEMG, and TA), and sometimes EEG signals. The number and type of recording signals are different dependent upon the study objective. Among them only three channels (airflow, PCA, and DEMG) are used in the automated segmentation system.

Digitized data is stored in a file which consists of the sequential order of each channel data with binary format, i.e., ch1, ch2,..., ch8,ch1,ch2,...ch8,..... Thus, after digitization, the data is separated into 8 different files of the same channel data. The airflow file is used for burst activity detection. Two files, PCA and DEMG data, are used as test

signals for boundary detection. The binary data are transferred to the NeXT workstation in the Neuroengineering laboratory through the UF Ethernet network, using File Transfer Protocol. After transferring the data, higher and lower 8 bits of data are swapped because the configuration of data format in the NeXT computer is not same as the digitized data format in the personal computer. The flow of data acquisition and transfer is illustrated in Figure 5.1.

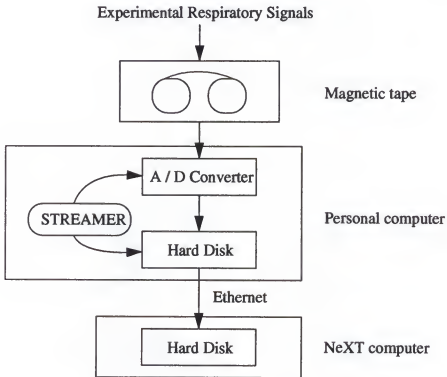


Figure 5.1. Flow of data acquisition and transfer

5.2 Evaluation Procedures

The system may find several boundary estimates in the transitional segment. The final boundary was determined as the most probable boundary for the given transitional segment. Thus, every transitional segment has only one final boundary estimate. This boundary is referred to as a "system-detected boundary" which will be used in the system evaluation. The segmentation system also produces the transitional segments such as the

coarse and the fine transitional segments. First, the way to evaluate the boundary is addressed.

The most reasonable evaluation for the system performance is to compare system-detected boundaries with human-detected boundaries. However, it is not guaranteed that the boundaries detected by a single EMGer are always reliable and correct because of the subjectiveness and intra-observer inconsistency. To reduce variations of the boundary detection accuracy, two EMGers were asked to choose the boundaries. This, on the other hand, creates the problem of inter-human variability, but gives more data to test the automated system.

The same recorded data which consisted of two channels of signals (PCA and DEMG) and their filtered signals were printed on paper and given to two EMGers. Since they know where the boundaries are, they were only asked to draw the boundaries on the filtered signals with reference to the original signals. After the EMGers marked the boundaries independently on the paper tracing, the human-detected boundaries were entered into the computer to quantify them in discrete time. The editing of boundary points was performed using the Event Edit Window of the automated segmentation system. All human-detected boundaries are grouped into four parts: PCA onset, DEMG onset, PCA cessation, and DEMG cessation. The human-detected boundaries can be displayed on the computer monitor to verify the boundary entry and can also be saved in a file for later comparison.

The system evaluation quantifies the accuracy of boundary detection in time. Thus, the boundaries in the transitional segments are calculated to get the relative timing difference between inter-EMGers, between EMGers and the system, and between average EMGers and the system. In other words, if we call A_i for one EMGer-detected boundary, B_i for another EMGer-detected boundary, and S_j for the system-detected final boundary, four types of differences can be computed as:

$$dAB_i = A_i - B_i$$

$$dAS_i = A_i - S_i$$

$$dBS_i = B_i - S_i$$

$$dABS_i = ((A_i + B_i)/2) - S_i.$$

The results are presented in two ways: an average difference between the EMGers, and an average difference between the EMGers and the system, i.e. the mean and its distribution of dAB_i , dAS_i , dBS_i , and $dABS_i$.

Another important evaluation is to check if the human-detected boundaries are located in the transitional segments. If the computer marks the transitional segments reliably, the EMGers only need to focus their attention on these segments, speeding up the analysis and eventually providing data reduction (only this data needs to be stored).

Five different EMG records which do not belong to the system training set are digitized and used to evaluate the system performance. The data for the system evaluation are shown in Table 5.1. Since one breath has four test cases such as onset and cessation of both PCA and DEMG signals, a total of 138 breaths correspond to 552 test cases.

Table 5.1. Number of test data to be used for system evaluation

Subject	Number of Breath	No. of onset point	No. of cessation point
pa104.dat	29	58	58
pa110.dat	27	54	54
pa114.dat	27	54	54
pa023.dat	29	58	58
pa136.dat	26	52	52
Total number	138	552	

5.3 Test Results

Results of the signal segmentation and boundary detection are presented graphically and statistically. The graphical presentation provides visualization of the boundary estimates, computer attributed probabilities and final boundary. It also delimits of the transitional segments where the boundaries are located. The statistical presentation gives numerical expression for both the duration of the transitional segment and boundary detection. Boundary detection is evaluated based on the detection accuracy in time. It is assessed based on the final boundary regardless of the number of boundaries and their probabilities in the transitional segments.

5.3.1 Graphical Presentation

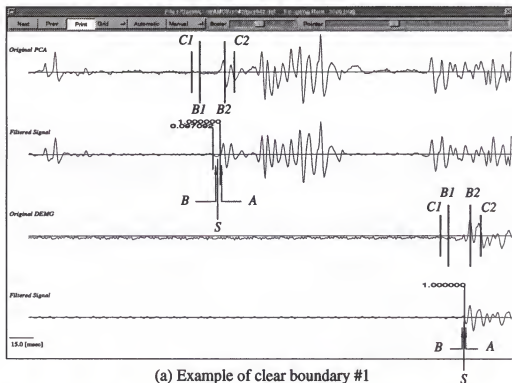
The boundaries with probabilities detected by the system are plotted along with those of EMGers. Some graphical examples are depicted in Figure 5.2. The figures show the transitional segments, system-detected boundaries with different probabilities, final boundaries, and EMGers-detected boundaries. The first channel of the figure is the raw PCA signal and the second is its filtered version. The third channel is the DEMG signal and the fourth is its filtered version. The numbers in the figure denote the boundary probabilities for the boundary estimates, which are determined by the neural network. For all figures in this chapter, unless otherwise specified, symbols, *A* and *B*, mean the boundaries detected by the EMGers. An interval between *C1* and *C2* (drawn by small bars) indicates the system defined coarse transitional segment, and an interval between *B1* and *B2* (drawn by large bars) indicates the fine transitional segment. The symbol, *S*, indicates the final boundary for a given transitional segment.

Examples for clear boundary conditions are shown in (a) and (b) of Figure 5.2. The system-detected and humans-detected boundaries are quite close. Another example of the boundary estimates with computer assigned probabilities is shown in (c). The human-detected boundaries, *A* and *B*, are close to the high probability boundary estimates.

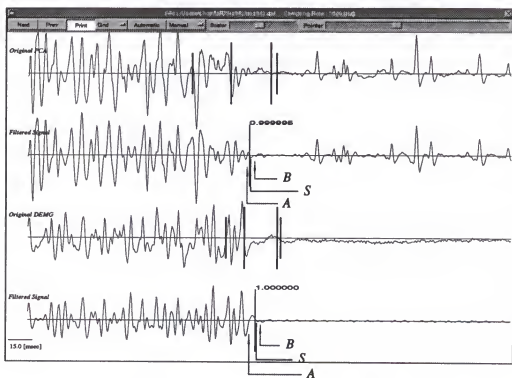
Examples for a fuzzy transitional segment are shown in (d). What is interesting in the last figure is that even though one human-detected boundary is relatively distant in time compared to the other, the system detects these two boundaries with high probability.

All these figures show that the human-detected boundaries, *A* and *B*, are inside the transitional segment and this is the rule for all our data set. Generally speaking, the proposed boundary detection method follows the human judgement of boundary determination for various patterns of signals. Thus, it can provide robust and reliable boundary detection.

Figure 5.3 shows two types of transitional segments: (a) for short transitional duration and (b) for long duration. Figures 5.4 and 5.5 show examples of larger differences between two humans, and between the system and the humans, respectively.

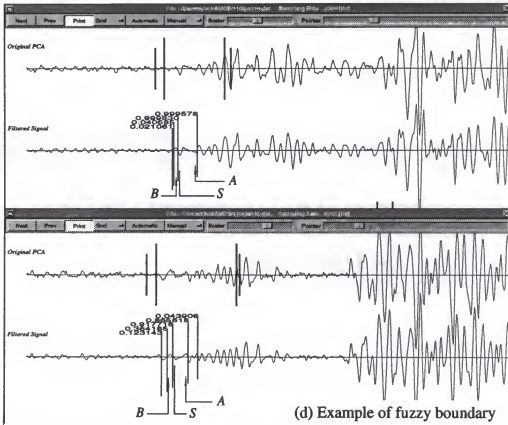
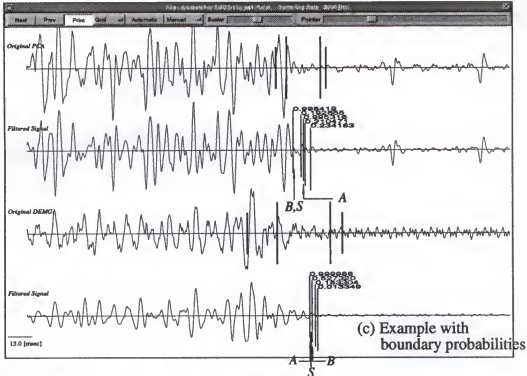


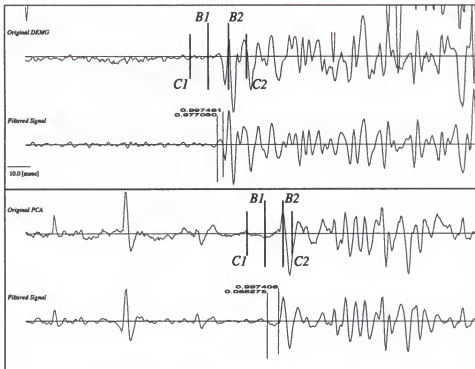
(a) Example of clear boundary #1



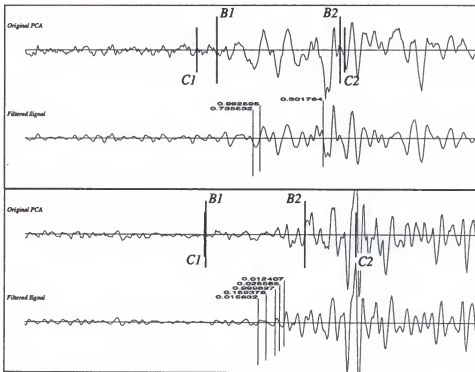
(b) Example of clear boundary #2

Figure 5.2. Some graphical results of boundary detection





(a) Examples of short transitional segments



(b) Examples of wide transitional segments

Figure 5.3. Transitional segments of variable duration

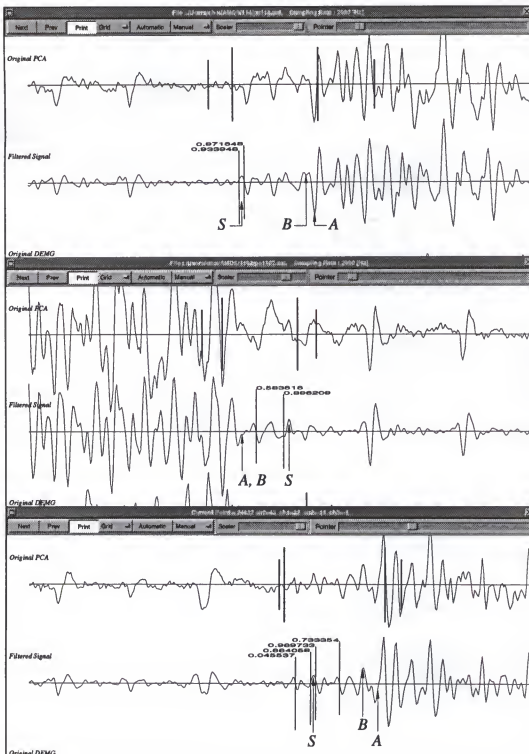


Figure 5.5. Examples of large boundary differences between the humans and system

5.3.2 Numerical Presentation

One of the problems encountered in basic statistics is the estimation of a parameter or some combination of parameters. An estimation of the population mean, μ , can be derived from the sample mean, \bar{x} . For sufficiently large sample sizes the sampling distribution of the sample mean is approximately normal. The confidence interval for the population mean is defined as

$$\bar{x} \pm k \frac{\sigma}{\sqrt{n}}$$

The parameter, σ , is the standard deviation of the sampled data and n is the sample size. The value of k is 1.645 for 90%, 1.960 for 95%, and 2.575 for 99% confidence interval.

Initial estimate for the sample size. The sample size will be influenced by the research objective and the type of parameters. The computed sample size is not a magical number to answer the minimum number required for an accurate statistical conclusion. It is an approximate figure for the number of needed samples. An equation to compute the sample size for estimating the average parameter, μ , is given by [Ma82]

$$n = \left(\frac{k}{B/\sigma} \right)^2$$

where k is obtained from a table of standard normal value for a chosen confidence coefficient. Remember that $k=1.645$ for confidence rate=0.9, $k=1.96$ for 0.95, and $k=2.575$ for 0.99. The "range" is defined as the difference between the largest and smallest possible measurements in a population. The standard deviation, σ , for a population can be estimated as one-fourth of the range, that is, $\sigma = \text{range} / 4$. This approximation has been known to be very accurate. Bound-on-error, B , in the above equation is a measure of the accuracy of a parameter estimate. It is the interval estimate. The sample size for this research can be computed as follows.

The maximum and minimum difference between human-detected boundaries and system-detected boundaries are assumed to be +80 points(=40[msec]) and -80 points. It is estimated based on a quite fuzzy transitional signal. The $\sigma = \text{range} / 4$ is $(80 - (-80)) / 4 =$

40. Now it is determined how accurately to estimate μ . In other words, we would like to estimate μ correctly within 10[msec], 20[msec], or 30[msec]. If we decide to estimate μ within 5[msec] which corresponds 10 points at 95% confidence rate, then $B=10$. Thus, the number of the sample size is $(1.96 / (10/40))^2 = 61.5$. It means that approximately 62 samples can provide an accurate estimation of the population mean with a 95% confidence interval and a 5[msec] bound-on-error. The minimum and maximum sample size of the test data records are 104 and 116 respectively, well above the required value.

5.3.2.1 Evaluation of the transitional segments

All boundaries and/or transitional segments detected by humans and the system are collected with time information. They were compared with each other based on timing differences. Table 5.2 shows the duration of the transitional segments and also shows how many human-detected boundaries are inside or outside the system-detected transitional segments.

Table 5.2. The duration of transitional segments and number of the humans-detected boundaries which are outside the segments

segment subject	coarse transitional segment [msec]	fine transitional segment [msec]	number outside coarse transitional segment	number outside fine transitional segment
pa104.dat	43.3±2.6	20.0±1.4	0 out of 116	2 out of 116
pa110.dat	46.6±2.7	24.4±1.8	0 out of 108	1 out of 108
pa114.dat	46.1±2.2	22.2±1.7	0 out of 108	3 out of 108
pa023.dat	48.3±2.4	23.3±1.6	0 out of 116	1 out of 116
pa136.dat	51.5±3.7	23.5±1.8	0 out of 104	0 out of 104
average	47.1±1.2	22.7±0.8	0 out of 552 (0%)	7 out of 552 (1.268%)

The values of the table indicate an average and standard deviation of the duration of the coarse and fine transitional segments at 95% confidence. An average of the total 552 test data is 47.1 ± 1.2 [msec] for the coarse transitional segment and 22.7 ± 0.8 [msec] for the fine transitional segment. There is no big difference in the transitional segment lengths among the five subjects, indicating similar signal characteristics across the subjects. If only the transitional segments are stored, the data reduction is as follows: 87.4% and 93.8% reduction in data for the coarse and fine transitional segments of PCA, 89.9% and 95.3% reduction for DEMG, and overall 88.6% and 94.5% reduction for the coarse and fine transitional segments.

The table also shows that two humans-detected boundaries are inside the coarse transitional segment. Only 1.268% (7 out of 552) are outside the fine transitional segment which is still a low ratio. In other words, we can pick up the boundaries within the minimum interval of 21.9[msec] and the maximum interval of 23.5[msec] with 98.714% correct detection ratio. There is only a 5% chance of making an error in the above statement. Seven boundary points outside the fine transitional segment are distributed across four subjects. Examples of human-detected boundaries which are outside the fine transitional segment are shown in Figure 5.6. The point, *H*, indicates the boundary detected by either human *A* or human *B*. It is outside the interval between *B1* and *B2* (fine segment), and inside the interval between *C1* and *C2* (coarse segment).

The distribution of the duration of the coarse and fine transitional segments is shown in Figure 5.7. Transitional segments, which range from 40[msec] to 60[msec] in the fine segment (*D1* in (a)) and 80[msec] to 100[msec] in the coarse segment (*D2* in (b)), correspond to more fuzzy transitional cases.

To test that the distribution of the transitional duration in the figure, say $f(n)$, is appropriately modelled as Gaussian, the following procedures are performed. First, mean and variance of the distribution are calculated as

$$m = \sum_n n \cdot f(n) \text{ and } \mu = \sum_n n^2 \cdot f(n) - m^2$$

An associated Gaussian function, $g(n)$, using m and μ is derived, and the difference between $f(n)$ and $g(n)$ is computed. The ratio of difference energy with respect to the $f(n)$ is calculated as

$$[f(n) - g(n)]^2 / (f(n))^2$$

The ratio is 0.1328 and 0.1527 for the fine and the coarse transitional segment respectively, indicating that the $f(n)$ can be modelled approximately (86.72% and 84.73%) as the normal Gaussian distribution.

The skewness characterizes the degree of asymmetry of a distribution around its mean. The usual definition is [Pr88]

$$Skew(x_1 \dots x_N) = \frac{1}{N} \sum_{i=1}^N \left[\frac{x_i - \bar{x}}{\sigma} \right]^3$$

where \bar{x} is the mean of the values x_1, \dots, x_N and $\sigma = \sigma(x_1 \dots x_N)$ is the distribution's standard deviation. A positive value of the skewness signifies a distribution with an asymmetric tail extending out towards a more positive x ; a negative value signifies a distribution whose tail extends out towards a more negative x (Figure 5.8). For the above equation to be meaningful, we need to have some idea of its standard deviation as an estimator of the skewness of the underlying distribution. For the idealized case of a normal (Gaussian) distribution, the standard deviation of above equation is approximately $\sqrt{6/N}$. The computed skewness is 0.983768 and 0.596234 for the fine and the coarse segment, compared to 0.2236 and 0.19365 for Gaussian. In real life, it is good practice to believe in the skewness only when it is several or many times as large as this. Thus, both segments can be stated as positively skewed and the distribution of the fine segment is more positively skewed.

The kurtosis measures the relative peakedness or flatness of a distribution. The conventional definition of the kurtosis is [Pr88]

$$Kurtosis = \left\{ \frac{1}{N} \sum_{i=1}^N \left[\frac{x_i - \bar{x}}{\sigma} \right]^4 \right\} - 3$$

where the -3 term makes the value zero for a normal distribution. The standard deviation of the above equation as an estimator of the kurtosis of an underlying normal distribution is $\sqrt{24/N}$. The computed kurtosis is -0.169607 and -0.798744 for the fine and the coarse segment, compared to 0.4472 and 0.3873 for Gaussian. Both segments are flatter than the Gaussian and the distribution of the coarse segment is flatter than the fine segment distribution.

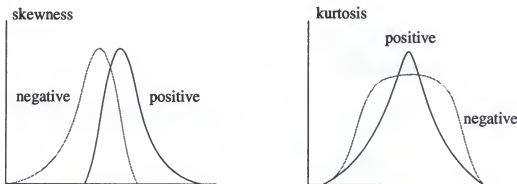


Figure 5.8. Distributions whose skewness and kurtosis are significantly different from a normal (Gaussian) distribution

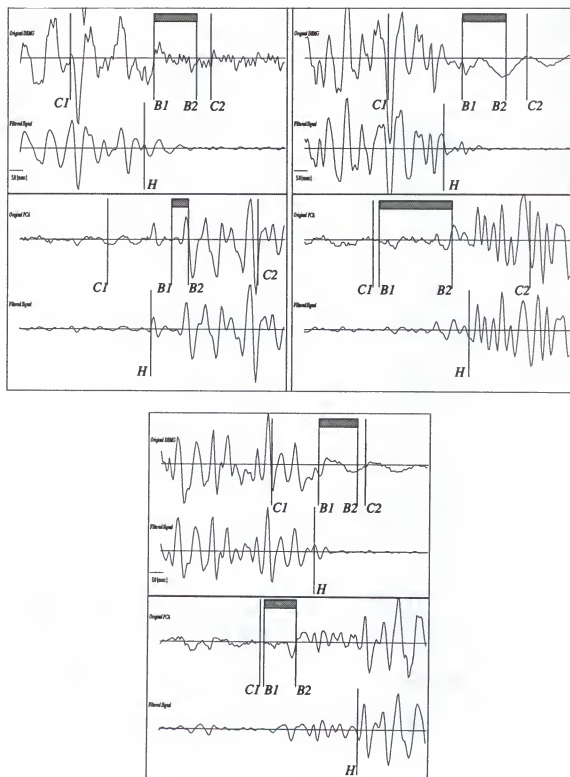


Figure 5.6. Examples of human-detected boundaries which are outside the fine transitional segment

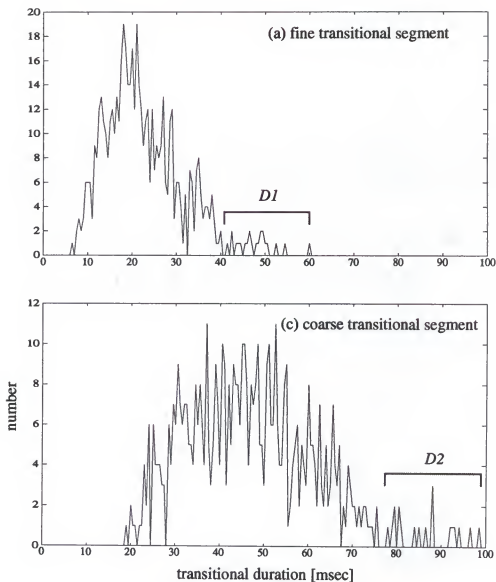


Figure 5.7. Distribution of the length of transitional segments

Table 5.3 is the same as Table 5.2 except that it presents the duration of transitional segments according to the transitional types, i.e., onset and cessational transition. The duration of the coarse transitional segment of onset is shorter than that of cessation, especially in the DEMG signal, which means that the signal is more variable during cessation than onset. However, the duration of the fine transitional segment of cessation is almost the same as that of the onset. For both transitional segments, the transitional interval of the DEMG is shorter than the PCA since the DEMG starts and ends more rapidly

compared to the PCA. This difference is more conspicuous in the onset transitional segment (51.5 versus 35.2[msec] for the coarse segment and 27.5 versus 17.8[msec] for the fine segment), reflecting the slow increase of the burst activity envelope of PCA onset compared to DEMG onset. For the fine transitional segment, DEMG onset shows the smallest duration while PCA onset is the lengthiest. It is interesting that most of the human-detected boundaries which are outside of the fine transitional segment are found in PCA onset and DEMG cessation, i.e., three PCA onset, one PCA cessation, and three DEMG cessation.

Table 5.3. The length of transitional segments and the number of humans-detected boundaries outside the segments based on the transitional types

transitional segment transitional type		coarse transitional segment [msec]	fine transitional segment [msec]	number outside coarse segment	number outside fine segment
onset	PCA	51.5±2.6	27.5±1.9	0 out of 138	3 out of 138
	DEMG	35.2±1.4	17.8±1.1	0 out of 138	0 out of 138
	average	43.4±1.8	22.6±1.2	0 out of 276 (0%)	3 out of 276 (1.09%)
cessation	PCA	52.7±2.5	23.6±1.3	0 out of 138	1 out of 138
	DEMG	49.0±1.9	21.7±1.1	0 out of 138	3 out of 138
	average	50.8±1.6	22.7±0.9	0 out of 276 (0%)	4 out of 276 (1.45%)

5.3.2.2 Boundary comparison between humans and the system

Four different groups between humans-detected boundaries and system-detected boundaries are compared. The average difference in milliseconds between two EMGers \overline{dAB}_i , average difference between an EMGer A and the system \overline{dAS}_i , average difference between an EMGer B and the system \overline{dBS}_i , and average difference between the mean of two

EMGers and the system \overline{dABS}_i , are presented for each subject and overall at a 95% confidence level. They are also presented by transitional types, that is, according to onset and cessational boundaries. Table 5.4 shows a summary of the boundary comparison between subjects and Table 5.5 shows the boundary comparison between transitional types.

Table 5.4. Comparison between four groups according to the subjects

comparison group subject	human A and human B \overline{dAB}_i [msec]	human A and system \overline{dAS}_i [msec]	human B and system \overline{dBS}_i [msec]	mean of human and system \overline{dABS}_i [msec]
pa104.dat	-0.4±1.0	-0.3±0.9	0.1±0.6	-0.1±0.6
pa110.dat	-0.6±1.1	-0.5±0.9	0.1±0.9	-0.2±0.7
pa114.dat	-1.2±1.0	0.9±1.7	2.1±1.3	1.5±1.4
pa023.dat	-1.5±0.6	-1.5±0.9	0.1±0.6	-0.7±0.7
pa136.dat	-0.8±1.0	-1.2±1.2	-0.4±1.1	-0.8±1.0
average	-0.9±0.4	-0.5±0.5	0.4±0.4	-0.06±0.4

The inter-human difference (\overline{dAB}_i) of the boundary detection is larger than other comparisons. The average differences among subjects in \overline{dAB}_i are consistent in time, i.e., human A-detected boundaries are ahead of human B's boundaries by 0.9[msec]. System-detected boundaries are generally detected after the human A's boundaries by 0.5[msec] and before the human B's boundaries by 0.4[msec]. However, the mean of two EMGers and system-detected boundaries compensate each other. Thus, \overline{dABS}_i shows the smallest value. On average, the system and the mean of the two EMGers are the closest in boundary detection, i.e., -0.06±0.4[msec]. The timing relationship between the EMGers and the system is shown Figure 5.9.

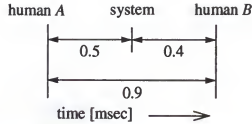


Figure 5.9. Timing relationship between humans and the system in terms of an average difference

Table 5.5. Comparison between four groups according to the transitional types

transitional type \ comparison group		human A and human B \overline{dAB}_i [msec]	human A and system \overline{dAS}_i [msec]	human B and system \overline{dBS}_i [msec]	mean of human and system \overline{dABS}_i [msec]
onset	PCA	0.3±1.0	2.9±1.3	2.6±1.2	2.7±1.2
	DEMG	1.7±0.5	1.6±0.5	-0.1±0.3	0.8±0.3
	average	1.0±0.6	2.3±0.7	1.2±0.6	1.8±0.6
cessation	PCA	-1.9±0.9	-2.8±1.0	-0.9±0.9	-1.8±0.9
	DEMG	-3.8±0.6	-3.8±0.6	0.02±0.4	-1.9±0.4
	average	-2.9±0.5	-3.3±0.6	-0.4±0.5	-1.9±0.5

In the comparison by transitional type, the average difference between human A and the system is the biggest for both onset (2.3[msec]) and cessation (3.3[msec]). For onset, \overline{dAB}_i shows the minimum average value (1.0[msec]) while for cessation, \overline{dBS}_i shows the minimum (0.4[msec]). It is interesting that for onset all means are positive, indicating that human B detects earlier than human A by 1.0[msec], and both EMGers detect the boundaries before the system by 2.3[msec] and 1.2[msec]. For cessation, all means are

negative, indicating that human *B* detects the boundaries after human *A* by 2.9[msec], and both EMGers detect boundaries after the system by 3.3[msec] and 0.4[msec]. This fact indicates that there is a systematic difference in measurement of the burst activity interval between humans and the system. That is, human *A* is stricter in the boundary detection than human *B*, leading to a slightly shorter burst activity duration by about 3.9 (1.0+2.9)[msec]. Humans are also stricter than the system, leading to a shorter interval by about 5.6 (2.3+3.3)[msec] for human *A* and the system, 1.6 (1.2+0.4)[msec] for human *B* and the system, and 3.7 (1.8+1.9)[msec] for the mean of humans and the system.

This interval difference is probably caused by the fact that, in boundary determination, the EMGers decide on the boundaries visually based on observation of overall background while the system detects the boundaries based on the background segment near the boundaries (specifically, the segment near the fine transitional segment) rather than the overall background. The level of the overall background is generally larger than the background level near the fine segment since it frequently contains tonic activity of large amplitude.

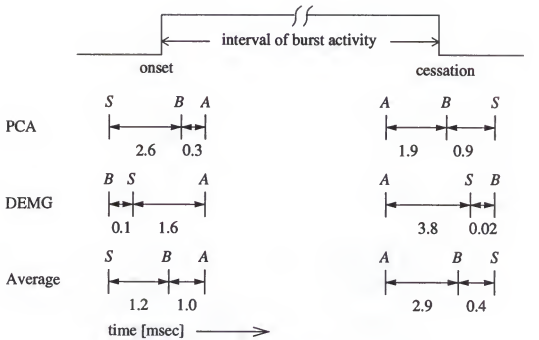
Based on Table 5.5, the system-detected boundaries are closer to the human *B* rather than human *A* for both onset and cessation boundary detection. Figure 5.10 shows the relative timing difference of boundary detection between groups, and between humans and the system.

Table 5.4 shows the mean value of boundary difference between groups. It does not provide the distribution of actual timing differences of boundaries. Figure 5.11 depicts a graphical distribution of the amount of timing difference for the four different groups, dAB_i , dAS_i , $dB S_i$ and $dABS_i$. The figure (a), dAB_i distribution, shows a little more widespread shape at mean value of -0.9[msec] than the others, i.e., dAS_i , $dB S_i$ and $dABS_i$. These figures illustrate that the difference distribution of human and system-detected boundaries are closer than that of two EMGers-detected boundaries. However, in terms of minimum and maximum difference, the system produces more numbers of large positive differences in

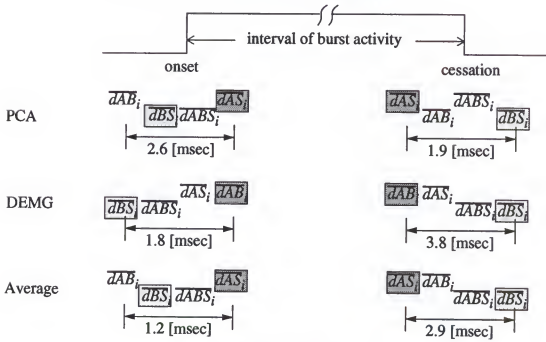
20[msec] through 30[msec] while showing fewer numbers of large negative differences in -20[msec] through -10[msec].

Since the figure (c), dBS_i distribution, shows the least dispersion at 0.4[msec], it can be said that the human B and the system are the closest in the boundary detection. The same happens for the comparison of mean value based on the transitional types, which is shown in Table 5.5. However, in terms of an average, the \overline{dABS}_i shows the least value (refer to Table 5.4) because the distribution of $dABS_i$ is more symmetric.

Overall, the variability between the system-detected boundaries and boundaries detected by either human was less than the variability between humans. However, since the absolute difference value between the EMGers, or between the humans and the system are so small, we can conclude that physiologically the differences between the four groups are actually same.



(a) between humans and system



(b) between four groups

Figure 5.10. Timing relationship of boundary detection between four groups

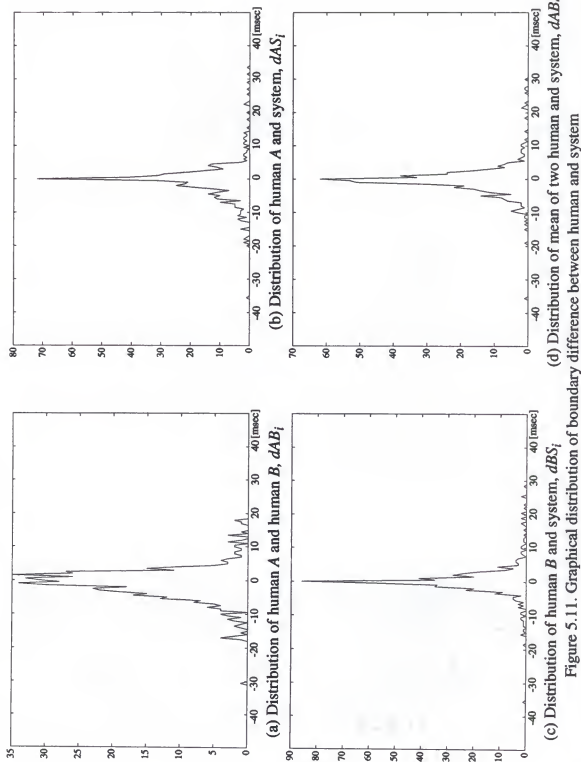


Figure 5.11. Graphical distribution of boundary difference between human and system

CHAPTER 6

CONCLUSION

The problem of signal segmentation and boundary detection (onset and cessation boundaries of the burst activity) of nonstationary respiratory EMG signals has been addressed. The limitation of the common method currently used has been outlined and a new approach of boundary detection with a new boundary presentation has been described.

In this chapter, the main ideas and contributions contained in this dissertation are described, and directions for further research in the respiratory EMG signal analysis are suggested, especially in the problem of detecting transitional boundaries.

6.1 Summary of Main Ideas

This dissertation describes the development and implementation of the multiresolution EMG segmentation model to determine boundaries for the breathing pattern of respiratory signals and to evaluate the model by designing a system. The new method of the automated signal segmentation was developed to improve detection accuracy of the boundaries in time. A new concept of boundary presentation was also proposed. Because of the signal fuzziness of the transitional phenomena, the determination of the transitional boundaries is not an easy problem in terms of physiological and signal viewpoints. That is, there may exist several probable boundary points with different degrees of certainty. Thus, outputs of boundary detection are assigned with probabilities.

The underlying methodology of the segmentation model concentrates on the problem of segmenting the signal and estimating transitional boundaries. Before processing EMG signal analysis, noise components are reduced using a bandpass filter to keep major frequencies of the burst activity and using an adaptive notch filter to reject electrical

interferences.

The conceptual approach of the model is performed through the multi-resolution analysis which narrows the duration of the signal segment targeted for in-depth analysis. In the signal segmentation, the burst activity of the EMG signal is recognized using multichannel information. The segment of the clear burst activity is discriminated from that of the background. Since the boundaries, either onset or cessation boundaries, are located between these two segments, the signal segmentation is further processed to derive the transitional segment. Here, two variable transitional segments are introduced: a coarse transitional segment of a relatively long interval which is guaranteed to contain the boundaries, and a fine transitional segment of a short interval which is determined tightly. For the clear and sharp transitional signal, the duration of the transitional segment is quite small compared to the fuzzy transitional signal. The presentation of the transitional segments is quite informative because the clinician only needs to focus his/her attention on these segments, speeding up the analysis and eventually providing data reduction.

All probable boundary candidates of half-waves are detected using the energy functions in the transitional segments. The pattern classification using an artificial neural network is utilized to derive the boundary discriminant. The discriminant function is computed by training the network through clear boundary examples. Of course, it is assumed that, based on proper network topology and suitable training data sets, the network produces an unbiased estimation. Thus, the boundary candidates are estimated through the neural network. Due to the property of the probability estimation of the network, the boundary estimate has a posterior probability. The boundary presentation with its probability is very useful in understanding the variable boundary phenomena and in determining the best boundary in circumstances of the unclear benchmark of the boundary definition. The final boundary point which can be considered as the most probable boundary for the given transitional segment is determined in the half-wave boundary estimate of the highest probability.

The overall functional flow of signal segmentation and boundary estimation supported by multi-channel information and multi-resolution approach is shown in Figure 6.1.

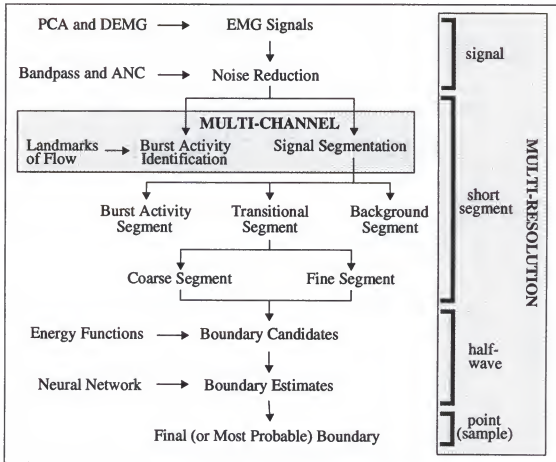


Figure 6.1. Functional flow of signal segmentation and boundary estimation

Two different efforts are explored to relieve obstacles that hamper the clinical application of EMG signal analysis. First, the general-purpose and computer-based signal analysis tool ("Signal Editor") was developed. The system is designed for the measurement of parameters and for displaying any combination of processed and original signals. In other words, algorithms can be tested graphically and numerically. Through the use of the window-based, menu-driven software package, the user can search, measure, save, and review portions of selected signals interactively and dynamically. Thus, the tool can be

used for the design and evaluation of signal processing algorithms which the user developed, for example, filters and waveform detection algorithms. Second, an automated segmentation system for the respiratory EMG signals was developed to evaluate the segmentation model and to partially help the EMGer in understanding the muscle signal analysis and boundary determination of respiratory EMG signals.

The automated segmentation system determines boundary estimates with probabilities, final boundaries, and transitional segments. Experimentation has shown that all human-detected boundaries are found in the coarse transitional segments and most of them (545 out of 552, i.e., 98.73%) are in the fine transitional segments. The average interval of the coarse and fine segments with a 95% confidence interval is 47.1 ± 1.2 [msec] and 22.7 ± 0.8 [msec] respectively. The system-detected final boundaries are compared with two EMGers(A and B)-detected boundaries to assess the system performance. Comparison results are -0.9 ± 0.4 [msec] between the humans, -0.5 ± 0.5 [msec] between human A and the system, 0.4 ± 0.4 [msec] between human B and the system, and -0.06 ± 0.4 [msec] between the mean of the humans and the system. Although the boundary differences between the humans and the system are smaller than those of the humans we can conclude that there are physiologically no differences between the groups because the timing difference is so small. Unfortunately, there is no available information to compare these results. This dissertation is the first trial in presenting numerical results for boundary detection.

The boundary of respiratory EMG signal was defined conceptually and has been implemented in some papers. For objective and accurate boundary detection using a computer, a more precise definition is required. In fact, the clear definition cannot be made due to the physiological phenomena. Considering that the transitional signal patterns are so diverse, integrated information extracted from clear and sufficient boundary examples can be an alternative expression of the boundary definition. In this sense, the neural net application for boundary detection makes the boundary definition concrete. An ensemble of the clear boundary examples can be regarded qualitatively as the true boundary. The

numerical information for the ensemble of boundaries is computed after training and distributed in the network parameters. Thus, boundary estimation through the neural net produces objective and correct detection in a consistent way.

6.2 Future Work

Both the signal segmentation and boundary estimation methods are performed by assuming that the background segment of a lengthy duration exists since the segmentation model uses the background information. The correct detection of the pure background segment can improve the system performance. The respiratory data records which are not included in this dissertation sometimes do not have sufficient intervals of the background. For the system developed in this study to be general, an adjustable background segment (or cumulative background information) may be required.

The neural network application for boundary estimation has been verified to be useful. However, the neural network using clear boundary examples may not be generalized to deal with all kinds of transitional signal patterns. For derivation of the general discriminant function, more examples including fuzzy boundary examples are required to reduce large differences between humans and the system. Thus, quantitative information of the boundary definition, especially for fuzzy signals, is helpful in collecting training data set. Another way is to develop signal features of the transitional pattern. In the dissertation, the training set of the network consists exclusively of amplitude information. The only amplitude measure may not be sufficient to include the signal characteristics of unstable background and diverse transitional signal shapes. The future research for the signal features of input pattern is necessary to cover any patterns of transitional signals.

A major difficulty in the automated segmentation system is that there are several system parameters to be determined, for example, in the determination of the background level, signal transformation, and the criteria for final boundary decision. In this research, all parameters are determined empirically based on observation of the signal. It means that

the system performance can be varied based on how well the parameters can be fixed. Thus, the more reliable method of parameter estimation is desirable to improve system performance.

The computer-based visualization tool developed here was proven to be useful in the design and evaluation of algorithms. However, the tool needs to be improved in its flexibility of algorithms implementation for more comfortable clinical applications. Sufficient and general resident algorithms, for which the parameters can be easily changeable, are required. The addition of new user-developing algorithms and an update of current algorithms are also needed to be handled in an easy manner. To enhance computational speed, the algorithms can utilize a DSP (Digital Signal Processing) chip. Another important issue in improvement is in the interface between the visualization tool and other softwares. For example, in the NeXT computer, the tool can communicate with Mathematica and Lotus Improv for further data processing.

REFERENCES

- [Au91] Austin, S., Zavaliagkos, G., Makhoul, J., and Schwartz, R., "A hybrid continuous speech recognition system using segmental neural nets with hidden Markov model," IEEE-SP Workshop on Neural Networks for Signal Processing, Princeton, NJ, 1991.
- [Ba85] Basmajian, J.V. and DeLuca, C.J., "Muscles alive; Their functions revealed by electromyography", Fifth edition, Williams & Wilkins, Baltimore, MD, 1985.
- [Bas75] Basmajian, J.V., Kukulka, M.G., and Takebe, K., "Biofeedback treatment of foot-drop after stroke compared with standard rehabilitation technique: Effects on voluntary control and strength," Arch. Phys. Med. Rehab., vol. 56, pp. 231-236, 1975.
- [Bas81] Basseville, M., "Edge detection using sequential methods for change in level. Part II: Sequential detection of change in mean," IEEE Trans. ASSP, vol. ASSP-29, no.1, pp. 32-50, 1981.
- [Bas83] Basseville, M. and Benveniste, A., "Sequential detection of abrupt changes in spectral characteristics of digital signals," IEEE Trans. on Inf. Theory, vol. IT-29, no. 5, pp. 709-724, 1983.
- [Bo77] Bohlin, T., "Analysis of EEG signals with changing spectra using a short word Kalman estimator," Math. Biosci., vol. 35, pp. 221-259, 1977.
- [Br90] Brenner, D., Doty, K.L., Moallem, M., Ragner, G., and Principe, J.C., "Development of a neural network based dynamic touch sensory system," Proc. Sensors Expo West, pp. 301A1-10, Long Beach, CA, 1990.
- [Bru84] Bruce, E.N., "Techniques in the life sciences," P4/I, Respiratory Physiology, P408/1-P408/17, Elsevier Scientific Publishers Ireland Ltd., Amsterdam, 1984.
- [Ca72] Carrie, J.R.G., "A hybrid computer technique for detecting sharp EEG transients," Electroenceph. Clin. Neurophysiol., vol. 32, pp. 336-338, 1972.
- [Ch88] Choi, H. G., "Comparison of smoothing algorithms for electroencephalographic signal processing," Master's thesis, Univ. of Florida, 1988.
- [Cla90] Clapp, D., "The NeXT bible," Simon & Schuster Inc., New York, 1990.
- [Cib90] Claire, N.R., Subbarao, W.V., and Duara, S., "Microcomputer aided analysis of respiratory muscle activity in premature newborns," Annual Int. Conf. of IEEE Eng. in Med. and Biology Society, vol. 12, no. 3, pp. 1220-1221, 1990.
- [De79] De Luca, C.J., "Physiology and mathematics of myoelectric signals," IEEE Trans. on Biomed. Eng., vol. BME-26, no. 6, pp. 313-325, 1979.

- [Du73] Duda, R. and Hart, P., "Pattern classification and scene analysis," Wiley, New York, 1973.
- [Ed87] Edelman, G.M.E., "Neural Darwinism," Basic Books, New York, 1987.
- [Eg89] Eggers, M.D. and Khoun, T.S., "Adaptive preprocessing of nonstationary signals," Technical Report 849, MIT Lincoln Lab., May 1989.
- [El90] El-Jaroudi, A. and Makhoul, J., "A new error criterion for posterior probability estimation with neural nets," IEEE Int. Joint Conf. on Neural Networks, vol. III, pp. 185-192, San Diego, CA, 1990.
- [Ge84] Gerber, A., Studer, R.M., De Figueiredo, R.J.P., and Moschytz, G.S., "A new framework and computer program for quantitative EMG signal analysis," IEEE Trans. on Biomed. Eng., vol. BME-31, no. 12, pp. 857-863, 1984.
- [Gi90] Gish, H., "A probabilistic approach to the understanding and training of neural network classifiers," IEEE Int. Conf. ASSP, vol.3, pp.1361-1364, 1990.
- [Gl77] Glover, J.R., "Adaptive noise cancelling applied to sinusoidal interferences," IEEE Trans. ASSP, vol. ASSP-25, no. 6, pp. 484-491, 1977.
- [Go88] Gorman, R.P. and Sejnowski, T.J., "Learned classification of sonar targets using a massively parallel network," IEEE Trans. ASSP., vol. 36, no. 7, pp. 1135-1140, 1988.
- [Got87] Gotman, J., "Computer analysis of EEG in epilepsy," in Clinical Applications of Computer Analysis of EEG and Other Neurophysiological Signals, EEG Handbook, vol. 2, pp. 171-200, Elsevier Scientific, Amsterdam, 1987.
- [Gr80] Gray, R.M., Buzo, A., Gray, A.H., and Matsuyama, Y., "Distortion measures for speech processing," IEEE Trans. ASSP, vol. 28, no. 4, pp. 367-376, 1980.
- [Gu83] Guiheneuc, P., Calamel, J., Doncarli, C., Gitton, D. and Michel, C., "Automatic detection of pattern recognition of single motor unit potentials in needle EMG," In Desmedt J.E. (Ed.), Computer-Aided Electromyography, Progress in Clinical Neurophysiology, vol. 10, pp. 73-127, 1983.
- [Gus78] Gustafson, D.E., Willsky, A.S., Wang, J.Y., Lancaster, M.C., and Triebwasser, J.H., "ECG/VCG rhythm diagnosis using statistical signal analysis, Part I: Identification of persistent rhythms," IEEE Trans. Biomed. Eng., vol. BME-25, no. 4, pp. 344-353, 1978.
- [Ha77] Hall, M., Oppenheim, A.V., and Willsky, A.S., "Time-varying parametric modelling of speech," in Proc. IEEE Conf. Decision and Control, pp. 1085-1091, New Orleans, LA, 1977.
- [Hu73] Huhta, J.C. and Webster, J.G., "60Hz interference in electrocardiography," IEEE Trans. Biomed. Eng., vol. BME-20, pp. 91-101, 1973.

- [Is81] Isaksson, A., Wennberg, A., and Zetterberg, L.H., "Computer analysis of EEG signals with parametric models," in Proc. IEEE, vol. 69, no. 4, pp. 451-461, 1981.
- [Ish80] Ishii, N., Sugimoto, H., Iwata, A., and Suzumura, N., "Computer classification of the EEG time series by Kullback information measure," Int. J. Syst. Sci., vol. 11, no. 6, pp. 677-688, 1980.
- [Ja82] Jacobsen, S.C., Knutti, D.F., Johnson, R.T., and Sears, H.H., "Development of the Utah artificial arm," IEEE Trans. Biomed. Eng., vol. BME-29, pp. 249-269, 1982.
- [Ki83] Kimura, J., "Electrodiagnosis in diseases of nerve and muscle: Principles and practice," Davis, PA, 1982.
- [Ko85] Koffler, D. and Gotman, J., "Automatic detection of spike-and-wave bursts in ambulatory EEG recordings," Electroenceph. Clin. Neurophysiol., vol. 61, pp. 165-180, 1985.
- [Ko88] Kosch, P.C., Hutchison, A.A., Wozniak, J.A., Carlo, W.A., and Stark, A.R., "Posterior cricoarytenoid and diaphragm activities during tidal breathing in neonates," J. Appl. Physiol., vol. 64, pp. 1968-1978, 1988.
- [La81] Lamel, L.F., Rabiner, L.R., Rosenberg, A.E. and Wilpon, J.G., "An improved endpoint detector for isolated word recognition," IEEE Trans. ASSP, vol. ASSP-29, no. 4, pp. 777-785, 1981.
- [Le82] LeFever, R.S. and De Luca, C.J., "A procedure for decomposing the myoelectric signal into its constituent action potentials. 1. Technique, theory and implementation," IEEE Trans. on Biomed. Eng. vol. BME-29, pp. 149-157, 1982.
- [Lia87] Lippmann, R.P., "An introduction to computing with neural nets," IEEE ASSP Magazine, vol. 4, no. 2, pp. 4-22, 1987.
- [Lib87] Lippman, R.P. and Gold, B., "Neural classifiers useful for speech recognition," in 1st IEEE Int. Conf. on Neural Networks, vol. 4, pp. 417, 1987.
- [Lo66] Lourenco, R.V., Cherniack, N.S., Malm, J.R., and Fishman, A.P., "Nervous output from the respiratory center during obstructed breathing," J. Appl. Physiol. vol. 21, pp. 527-533, 1966.
- [Lu84] Luenberger, D., "Linear and nonlinear programming," Addition-Wesley, Reading, MA, 1984.
- [Ma91] Makhoul, H., "Pattern recognition properties of neural networks," Neural Networks for Signal Processing, Proc. of the 1991 IEEE Workshop, pp. 173-187, 1991.

- [Mo88] "Monarch," Software manual, The Athena Group Inc., Gainesville, FL, 1988.
- [Ne90] Neeman, H., McCall, W., Plesh, O., and Bishop, B., "Analysis of jaw movement and masticatory muscle activity," *Computer Methods and Programs in Biomedicine*, vol. 31, pp. 19-32, 1990.
- [Ne91] "NeXT on Campus," *Magazine*, vol. 2, issue 2, NeXT Computer Inc., Redwood City, CA, Winter 1991.
- [Op75] Oppenheim, A.V. and Schafer, R.W., "Digital signal processing," Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [Pa89] Pao, Y.H., "Adaptive pattern recognition and neural networks," Addison-Wesley, Reading, MA, 1989.
- [Pa90] Park, S-H, Principe, J.C., Smith, J.R., and Reid, A.S., "TDAT-Time domain analysis tool for EEG Analysis," *IEEE Trans. on Biomed. Eng.*, vol. 37, no. 8, pp. 803-811, 1990.
- [Par72] Parks, T.W. and McClellan, J.H., "Chebyshev approximation for nonrecursive digital filters with linear phase," *IEEE Trans. Circuit Theory*, vol. CT-19, pp. 189-194, 1972.
- [Pr88] Press, W. H., Flannery, B. P., Teukolsky, S. L., and Vetterling, W. T., "Numerical recipes in C," *The Art of Scientific Computing*, Cambridge University Press, Cambridge, MA, 1988.
- [Pri85] Principe, J.C. and Smith, J.R., "Automatic detection of spike-and-wave burst," In: Gotman J, Ives J.R., and Gloor P. (Eds.), *Long-Term Monitoring in Epilepsy. Electroenceph clin. Neurophysiol.*, Suppl. 37, pp. 115-131, Elsevier, Amsterdam, 1985.
- [Pri89] Principe, J.C. and Tome, A.M.P., "Performance and training strategies in feedforward neural networks: An application to sleep scoring," *Proc. Int. Conf. on Neural Networks*, vol. 1, pp. 341-346, 1989.
- [Ra78] Rabiner, L.R. and Schafer, L.W., "Digital processing of speech signals," Prentice-Hall, Englewood Cliffs, NJ, 1978.
- [Ro62] Rosenblatt, F., "Principles of neuro dynamics: Perceptrons and the theory of brain mechanisms," Spartan, New York, 1962.
- [Ru86] Rumelhart, D., Hinton, G., and Williams, R., "Learning representations by error propagation," in *parallel distributed processing: Explorations in the microstructure of cognition*, D. Rumelhart and J. McClelland (eds.), vol. 1, pp. 318-362, MIT press, Cambridge, MA, 1986.
- [Sa89] Savoji, M.H., "A robust algorithm for accurate endpoints of speech signals," *Speech Communication* 8, pp. 45-60, North-Holland, 1989.

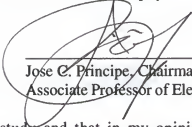
- [Sh80] Shensha, M.J., "Non-Wiener solutions of the adaptive noise canceller with a noisy reference," IEEE Trans. ASSP, vol. ASSP-28, pp. 468, 1980.
- [Si82] Siegel, L.J. and Bessey, A.C., "Voiced/unvoiced/mixed excitation classification of speech," IEEE Trans. ASSP, vol. ASSP-29, no. 3, pp. 451-460, 1982.
- [Sm89] Smith, R.P., "A simplified algorithm for identifying wave endpoints in digitized intrauterine pressure waveform data", Biomed. Instru. & Tech., pp. 298-300, 1989.
- [St86] Stalberg F. and Boon K., "Electromyography," in Handbook of Electroencephalography and Clinical Neurophysiology, Revised Series vol. 2, Elsevier, Amsterdam, 1986.
- [To74] Tou, J.T. and Gonzalez, R.C., "Pattern recognition principles," Addison-Wesley, Reading, MA, 1974.
- [Vo81] Vodovnik, L., Bajd, T., Kralj, A., Gracanin, F., and Strojnik, P., "Functional electrical stimulation for control of locomotor systems," Critic Review Bioeng., vol. 6, pp. 63-131, 1981.
- [Wh89] White, H., "Learning in artificial neural networks: A statistical perspective," Neural Computation, vol. 1, no. 4, pp. 425-464, 1989.
- [Wi66] Widrow, B., "Adaptive filters I: Fundamentals," Stanford Electron. Labs., Stanford Univ., Rep. SEL-66-126, CA, 1966.
- [Wi85] Widrow, B. and Sterns, S. D., "Adaptive signal processing," Prentice-Hall, Englewood Cliffs, NJ, 1985.
- [Wil84] Wilpon, J.G., Rabiner, L.R., and Martin, T., "An improved word-detection algorithm for telephone-quality speech incorporating both syntactic and semantic constraints," AT&T Bell Lab. Tech. Journal, vol. 63, no. 3, pp. 479-498, 1984.
- [Wo38] Wold, H., "A study in the analysis of stationary time series," Almquist and Wicksells, Uppsala, Sweden, 1938.
- [Ya82] Yamada, Y., Ash, J.L., Ash, G.M., and Ash Jr., M.M., "A measuring system for EMG silent period using a microcomputer," IEEE Trans. on Biomed. Eng., vol. BME-29, no. 11, pp. 713-718, 1982.
- [Ya80] Yamada, Y., Ishioka, K., and Ash Jr., M.M., "An automated measuring system for EMG silent period," IEEE Trans. on Biomed. Eng., vol. BME-27, no. 7, pp. 410-413, 1980.

BIOGRAPHICAL SKETCH

Haan-Go Choi was born on May 3, 1958, in Korea. In 1979, he graduated from Kyungpook National University in Taegu, Korea, with a bachelor's degree from the Department of Electrical Engineering. From 1979 to 1986, he was a member of the technical staff of the Agency for Defense Development, Daejeon, Korea.

In 1986, he enrolled in the Electrical Engineering Department of the University of Florida, Gainesville, where he received his Master of Science degree in 1988. Since then, he has worked on various topics in digital signal processing and computer application at the Computational Neuroengineering Laboratory to continue his Ph.D. degree.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Jose C. Principe, Chairman
Associate Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Jack R. Smith
Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



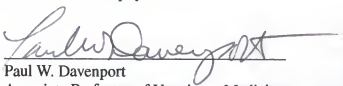
John Staudhammer
Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



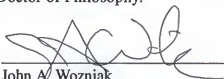
Alastair A. Hutchison
Associate Professor of Pediatrics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Paul W. Davenport
Associate Professor of Veterinary Medicine

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



John A. Wozniak
Assistant Professor of Pediatrics

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December 1992



Winfred M. Phillips
Dean, College of Engineering

Madelyn M. Lockhart
Dean, Graduate School